

# TOOLKIT API

**Dresden University of Technology**

Copyright 2010

---

Package  
**tud.iir.classification**

# tud.iir.classification

## Class Categories

```

java.lang.Object
  |-- java.util.AbstractCollection
        |-- java.util.AbstractList
              |-- java.util.ArrayList
                    |-- tud.iir.classification.Categories
  
```

### All Implemented Interfaces:

java.io.Serializable, java.util.Collection, java.util.List, java.io.Serializable, java.lang.Cloneable, java.util.RandomAccess, java.util.List

```

public class Categories
extends java.util.ArrayList
implements java.util.List, java.util.RandomAccess, java.lang.Cloneable,
java.io.Serializable, java.util.List, java.util.Collection, java.io.Serializable
  
```

An ArrayList of categories.

### Author:

David Urbansky

## Constructors

### Categories

```
public Categories()
```

## Methods

### contains

```
public boolean contains(java.lang.Object obj)
```

Check whether ArrayList contains obj.

#### Returns:

True if the obj is contained, false otherwise.

### containsCategoryName

```
public boolean containsCategoryName(java.lang.String categoryName)
```

### add

```
public boolean add(Category category)
```

## addAll

```
public boolean addAll(java.util.Collection c)
```

---

## getCategoryByName

```
public Category getCategoryByName(java.lang.String categoryName)
```

Get a certain category from the list.

**Parameters:**

categoryName

**Returns:**

category

---

## calculatePriors

```
public void calculatePriors()
```

After the learning phase, each category has a frequency. The ratio of frequency to total number of documents will be used to calculate the priors.

**Parameters:**

totalDocuments - The total number of documents having a category assigned.

# tud.iir.classification

## Class Category

java.lang.Object

└--tud.iir.classification.Category

All Implemented Interfaces:

java.io.Serializable

```
public class Category
  extends java.lang.Object
  implements java.io.Serializable
```

A category has a name and a relevance for certain resource.

**Author:**

David Urbansky

## Constructors

### Category

```
public Category(java.lang.String name)
```

## Methods

### getName

```
public java.lang.String getName()
```

### setName

```
public void setName(java.lang.String name)
```

### getFrequency

```
public int getFrequency()
```

### increaseFrequency

```
public void increaseFrequency()
```

(continued from last page)

---

## decreaseFrequency

```
public void decreaseFrequency()
```

---

## getPrior

```
public double getPrior()
```

The prior probability of this category. Set after learning.

**Returns:**

The prior probability of this category.

---

## setIndexedPrior

```
public void setIndexedPrior(double prior)
```

The prior can be indexed and read from the index. Instead of calculating it via `Categories.calculatePriors()`, it can be set using this method.

**Parameters:**

`prior`

---

## calculatePrior

```
public void calculatePrior(int totalDocuments)
```

Calculates the prior for this category, which is the ratio between this categories frequency to all documents in the corpus.

**Parameters:**

`totalDocuments` - The count of total documents on this corpus.

---

## isMainCategory

```
public boolean isMainCategory()
```

---

## setMainCategory

```
public void setMainCategory(boolean mainCategory)
```

---

## getClassType

```
public int getClassType()
```

---

## setClassType

```
public void setClassType(int classType)
```

(continued from last page)

---

## equals

```
public boolean equals(java.lang.Object obj)
```

Equality is checked by category name.

---

## toString

```
public java.lang.String toString()
```

---

## setTestSetWeight

```
public void setTestSetWeight(double testSetWeight)
```

---

## getTestSetWeight

```
public double getTestSetWeight()
```

---

## increaseTotalTermWeight

```
public void increaseTotalTermWeight(double totalTermWeight)
```

---

## getTotalTermWeight

```
public double getTotalTermWeight()
```

## tud.iir.classification Class CategoryEntries

```
java.lang.Object
├-- java.util.AbstractCollection
│   ├── java.util.AbstractList
│   │   ├── java.util.ArrayList
│   │   └── tud.iir.classification.CategoryEntries
```

### All Implemented Interfaces:

java.io.Serializable, java.util.Collection, java.util.List, java.io.Serializable, java.lang.Cloneable, java.util.RandomAccess, java.util.List

---

```
public class CategoryEntries
extends java.util.ArrayList
implements java.util.List, java.util.RandomAccess, java.lang.Cloneable,
java.io.Serializable, java.util.List, java.util.Collection, java.io.Serializable
```

Hold a number of category entries. For example, a word could have a list of relevant categories attached. Each category has a certain relevance for the word which is expressed in the CategoryEntry.

### Author:

David Urbansky

---

## Constructors

### CategoryEntries

```
public CategoryEntries()
```

---

## Methods

### isRelevancesUpToDate

```
public boolean isRelevancesUpToDate()
```

---

### setRelevancesUpToDate

```
public void setRelevancesUpToDate(boolean relevancesUpToDate)
```

---

### getCategoryEntry

```
public CategoryEntry getCategoryEntry(Category category)
```

---



(continued from last page)

## getCategoryEntry

```
public CategoryEntry getCategoryEntry(java.lang.String categoryName)
```

---

## setRelevancesInPercent

```
public void setRelevancesInPercent(boolean relevancesInPercent)
```

---

## transformRelevancesInPercent

```
public void transformRelevancesInPercent(boolean spread)
```

This method calculates the percentage for every category in the ArrayList. The sum of percentages of all categories must be 100% (+-1% round).

---

## add

```
public boolean add(CategoryEntry e)
```

---

## addAll

```
public boolean addAll(java.util.Collection c)
```

---

## calculateRelativeRelevances

```
public void calculateRelativeRelevances()
```

The relevance for a category entry is a sum of absolute relevance scores so far. To normalize the relevance to a value between 0 and 1 we need to divide it by the total absolute relevances of all category entries that are in the same category entries group.

---

## sortByRelevance

```
public void sortByRelevance()
```

---

## getMostLikelyCategoryEntry

```
public CategoryEntry getMostLikelyCategoryEntry()
```

---

## getTermWeight

```
public double getTermWeight(Category category)
```

Get the percentage of all absolute term weights for all category entries in the given category. The percentage tells what ratio of term weights were relevant for the given category in this entry set.

**Parameters:**

(continued from last page)

category - The category entry.

**Returns:**

The percentage.

---

## hasEntryWithCategory

```
public boolean hasEntryWithCategory(Category category)
```

# tud.iir.classification

## Class CategoryEntry

java.lang.Object

└--tud.iir.classification.CategoryEntry

All Implemented Interfaces:

java.io.Serializable

---

```
public class CategoryEntry
    extends java.lang.Object
    implements java.io.Serializable
```

Hold information about how relevant a category is.

Author:

David Urbansky

---

## Fields

### bayesRelevance

```
public double bayesRelevance
```

## Constructors

### CategoryEntry

```
public CategoryEntry(CategoryEntries categoryEntries,
                    Category category,
                    double absoluteRelevance)
```

## Methods

### getCategory

```
public Category getCategory()
```

---

### setCategory

```
public void setCategory(Category category)
```

---

### getRelevance

```
public double getRelevance()
```

(continued from last page)

---

## multAbsRel

```
public void multAbsRel(double factor)
```

---

## getAbsoluteRelevance

```
public double getAbsoluteRelevance()
```

---

## addAbsoluteRelevance

```
public void addAbsoluteRelevance(double value)
```

---

## getCategoryEntries

```
public CategoryEntries getCategoryEntries()
```

---

## setCategoryEntries

```
public void setCategoryEntries(CategoryEntries categoryEntries)
```

---

## toString

```
public java.lang.String toString()
```

# tud.iir.classification Class Classifier

java.lang.Object

└tud.iir.classification.Classifier

Direct Known Subclasses:

[WhereClassifier](#), [SnippetClassifier](#), [AnswerClassifier](#), [MIOClassifier](#), [EntityClassifier](#)

```
public class Classifier
extends java.lang.Object
```

## Fields

### BAYES\_NET

```
public static final int BAYES_NET
```

Constant value: 1

### LINEAR\_REGRESSION

```
public static final int LINEAR_REGRESSION
```

Constant value: 2

### SVM

```
public static final int SVM
```

Constant value: 3

### NEURAL\_NETWORK

```
public static final int NEURAL_NETWORK
```

Constant value: 4

### SVM2

```
public static final int SVM2
```

Constant value: 5

## Constructors

(continued from last page)

## Classifier

```
public Classifier(int type)
```

## Methods

### trainClassifier

```
public void trainClassifier(java.lang.String filePath)
```

Train a classifier with data from a file. The file must be structured as follows: Each line is one object in an n-dimensional vector space. All features and the class must be numeric.  
f1;f2;...;fn;class

**Parameters:**

filePath - The path that points to the training file.

### testClassifier

```
public void testClassifier(int conceptID)
```

Test a classifier with the samples save in the database. The classifier is tested on a concept level.

**Parameters:**

conceptID - The id of the concept for which the classifier should be tested.

featureString - The SQL query string with the desired features to test the classifier.

### testClassifier

```
public void testClassifier(java.lang.String filePath)
```

### readFeatureObjects

```
public java.util.ArrayList readFeatureObjects(int conceptID,  
java.sql.PreparedStatement featureQuery)
```

### readFeatureObjects

```
public java.util.ArrayList readFeatureObjects(java.lang.String filePath)
```

Load feature objects from a file.

**Parameters:**

filePath - The file with the training data.

**Returns:**

A list with the feature objects.

### getFvWekaAttributes

```
public FastVector getFvWekaAttributes()
```

(continued from last page)

---

## setFvWekaAttributes

```
public void setFvWekaAttributes(FastVector fvWekaAttributes)
```

---

## getPsFeatureStatement

```
public java.sql.PreparedStatement getPsFeatureStatement()
```

---

## setPsFeatureStatement

```
public void setPsFeatureStatement(java.sql.PreparedStatement psFeatureStatement)
```

---

## getPsClassificationStatementConcept

```
public java.sql.PreparedStatement getPsClassificationStatementConcept()
```

---

## setPsClassificationStatementConcept

```
public void setPsClassificationStatementConcept(java.sql.PreparedStatement psClassificationStatement)
```

---

## getPsClassificationStatementEntity

```
public java.sql.PreparedStatement getPsClassificationStatementEntity()
```

---

## setPsClassificationStatementEntity

```
public void setPsClassificationStatementEntity(java.sql.PreparedStatement psClassificationStatementEntity)
```

---

## getTrainingSet

```
public Instances getTrainingSet()
```

---

## setTrainingSet

```
public void setTrainingSet(Instances trainingSet)
```

## isDiscrete

```
public boolean isDiscrete()
```

---

## setDiscrete

```
public final void setDiscrete(boolean discrete)
```

---

## getTrainingObjects

```
public java.util.ArrayList getTrainingObjects()
```

---

## setTrainingObjects

```
public void setTrainingObjects(java.util.ArrayList trainingObjects)
```

---

## getChosenClassifier

```
public final int getChosenClassifier()
```

---

## getChosenClassifierName

```
public java.lang.String getChosenClassifierName()
```

---

## setChosenClassifier

```
public final void setChosenClassifier(int chosenClassifier)
```

---

## isNominalClass

```
public boolean isNominalClass()
```

---

## setNominalClass

```
public void setNominalClass(boolean nominalClass)
```

---

## getClassifier

```
public weka.classifiers.Classifier getClassifier()
```

---



(continued from last page)

---

## setClassifier

```
public void setClassifier(weka.classifiers.Classifier classifier)
```

---

## getEvaluation

```
public Evaluation getEvaluation()
```

---

## setEvaluation

```
public void setEvaluation(Evaluation evaluation)
```

---

## getRMSE

```
public double getRMSE()
```

---

## getFeatureCombination

```
public java.lang.String getFeatureCombination()
```

---

## classifyBinary

```
public boolean classifyBinary(FeatureObject fo,  
    boolean output)
```

Classify a feature object binary.

**Parameters:**

fo - The feature object.

**Returns:**

true if positive, false otherwise

---

## classifySoft

```
public double[] classifySoft(FeatureObject fo)
```

Classify an object soft, return distribution. Index 0 is the probability that it is positive, index 1 that it is negative.

**Parameters:**

fo

**Returns:**

(continued from last page)

## main

```
public static void main(java.lang.String[] args)
```

# tud.iir.classification

## Class Dictionary

```

java.lang.Object
  |
  +- java.util.AbstractMap
      |
      +- java.util.HashMap
          |
          +- tud.iir.classification.Dictionary
  
```

### All Implemented Interfaces:

java.io.Serializable, java.util.Map, java.io.Serializable, java.lang.Cloneable, java.util.Map, java.io.Serializable

```

public class Dictionary
extends java.util.HashMap
implements java.util.Map, java.lang.Cloneable, java.io.Serializable, java.util.Map,
java.io.Serializable
  
```

A dictionary holds a list of words with their probabilities/scores of belonging to certain categories. Word Category1 ... CategoryN test 0.1 0.3 ...

### Author:

David Urbansky

## Fields

### hierarchyRootNode

```
public tud.iir.helper.TreeNode hierarchyRootNode
```

the hierarchy of categories (for hierarchical classification)

### DB\_INDEX\_FAST

```
public static final int DB_INDEX_FAST
```

save dictionary in a database all in one table  
Constant value: 1

### DB\_INDEX\_NORMALIZED

```
public static final int DB_INDEX_NORMALIZED
```

save dictionary in a database, normalized in three tables (slower than using one table)  
Constant value: 2

### LUCENE\_INDEX

```
public static final int LUCENE_INDEX
```

save dictionary on disk in Lucene index  
Constant value: 3

(continued from last page)

## DB\_MYSQL

```
public static final int DB_MYSQL
```

use client server mysql  
Constant value: 1

## DB\_H2

```
public static final int DB_H2
```

use embedded h2  
Constant value: 2

## Constructors

### Dictionary

```
public Dictionary(java.lang.String name,  
                  int classType)
```

### Dictionary

```
public Dictionary(java.lang.String name,  
                  int classType,  
                  int indexType,  
                  int databaseType)
```

## Methods

### useIndex

```
public void useIndex()
```

Open or create an index. Either in database or on a Lucene index on disk. The index is then ready to be read or written.

#### Parameters:

`classType` - The class type distinguishes certain indexes. There can be several indexes with the same name but only with different class types.

### closeIndexWriter

```
public void closeIndexWriter()
```

### emptyIndex

```
public void emptyIndex()
```

(continued from last page)

---

## useMemory

```
public void useMemory()
```

---

## isUseIndex

```
public boolean isUseIndex()
```

---

## setMainCategories

```
public void setMainCategories(Categories categories)
```

In hierarchical classification mode, the root category is the main category. For evaluation purposes we need to tell the dictionary which categories are main categories.

### Parameters:

categories - Categories of which some are main categories.

---

## updateWord

```
public CategoryEntries updateWord(Term word,  
    Category category,  
    double value)
```

---

## updateWord

```
public CategoryEntries updateWord(Term word,  
    java.lang.String categoryName,  
    double value)
```

---

## updateWCM

```
public void updateWCM(Term[] terms)
```

Update the word correlation matrix.

### Parameters:

terms - A set of terms that co-occurred

---

## getMostLikelyCategoryEntry

```
public CategoryEntry getMostLikelyCategoryEntry(java.lang.String word,  
    double minimumScore)
```

Get the best matching category for a given word.

### Parameters:

word - The word to be looked up.

minimumScore - The minimum score required to return the category.

### Returns:

(continued from last page)

The category name that the word is most likely associated with.

---

## getMostLikelyCategoryEntry

```
public CategoryEntry getMostLikelyCategoryEntry(java.lang.String[] words,  
double minimumScore)
```

---

## getCategoryEntries

```
public CategoryEntries getCategoryEntries(java.lang.String word,  
double minimumScore)
```

---

## getCategoryEntries

```
public CategoryEntries getCategoryEntries(java.lang.String[] words)
```

Get a list of categories that can be associated with the list of words.

### Parameters:

words - A list of words.

### Returns:

categories The categories the words belong to.

---

## getNumberOfDocuments

```
public int getNumberOfDocuments()
```

---

## setNumberOfDocuments

```
public void setNumberOfDocuments(int numberOfDocuments)
```

---

## increaseNumberOfDocuments

```
public void increaseNumberOfDocuments()
```

---

## saveAsCSV

```
public void saveAsCSV()
```

Save the constructed context map to a csv file.

---

## calculateCategoryPriors

```
public void calculateCategoryPriors()
```

## index

```
public void index(boolean deleteIndexFirst)
```

Write the complete dictionary to an index.

**Parameters:**

`indexPath` - The path of the index.

---

## serialize

```
public void serialize(java.lang.String indexPath,  
    boolean indexFirst,  
    boolean deleteIndexFirst)
```

Serialize the dictionary but without the actual entries. They can be retrieved from the index.

**Parameters:**

`indexPath`  
`indexFirst`

---

## get

```
public CategoryEntries get(Term term)
```

Get a list of category entries for the given term.

**Parameters:**

`term` - A term might be a word or any other sequence of characters.

**Returns:**

A list of category entries.

---

## getName

```
public java.lang.String getName()
```

---

## setName

```
public void setName(java.lang.String name)
```

---

## getCategories

```
public Categories getCategories()
```

---

## setCategories

```
public void setCategories(Categories categories)
```

---

---

## isReadFromIndexForUpdate

```
public boolean isReadFromIndexForUpdate()
```

---

---

## setReadFromIndexForUpdate

```
public void setReadFromIndexForUpdate(boolean readFromIndexForUpdate)
```

---

---

## toString

```
public java.lang.String toString()
```

---

---

## setClassType

```
public void setClassType(int classType)
```

---

---

## getClassType

```
public int getClassType()
```

---

---

## setIndexType

```
public void setIndexType(int indexType)
```

---

---

## getIndexType

```
public int getIndexType()
```

---

---

## getIndexPath

```
public java.lang.String getIndexPath()
```

---

---

## setIndexPath

```
public void setIndexPath(java.lang.String indexPath)
```

---

---

## setDatabaseType

```
public void setDatabaseType(int databaseType)
```

---



(continued from last page)

---

## getDatabaseType

```
public int getDatabaseType()
```

---

## setWcm

```
public void setWcm(WordCorrelationMatrix wcm)
```

---

## getWcm

```
public WordCorrelationMatrix getWcm()
```

# tud.iir.classification

## Class FastWordCorrelationMatrix

java.lang.Object

└- [tud.iir.classification.WordCorrelationMatrix](#)  
└- **tud.iir.classification.FastWordCorrelationMatrix**

All Implemented Interfaces:  
java.io.Serializable

---

```
public class FastWordCorrelationMatrix  
extends WordCorrelationMatrix
```

This implementation is about twice as fast as the [WordCorrelationMatrix](#), by using nested HashMaps to accelerate the look up of correlations, but therefor also consumes twice as much memory.

**Author:**  
Philipp Katz

---

## Constructors

### FastWordCorrelationMatrix

```
public FastWordCorrelationMatrix()
```

## Methods

### getCorrelation

```
public WordCorrelation getCorrelation(java.lang.String word1,  
    java.lang.String word2)
```

---

### getCorrelations

```
public java.util.Set getCorrelations()
```

Return all correlation pairs.

---

### getCorrelations

```
public java.util.List getCorrelations(java.lang.String word,  
    int minCooccurrences)
```

# tud.iir.classification

## Class FeatureEvaluator

```
java.lang.Object
├-- tud.iir.classification.FeatureEvaluator
```

```
public class FeatureEvaluator
    extends java.lang.Object
```

The FeatureEvaluator can be used to determine the value of features for different classifiers. Different combinations are tested with a training and a testing set. All features must be available from the database and it must be possible to determine using SQL.

**Author:**  
David Urbansky

## Constructors

### FeatureEvaluator

```
public FeatureEvaluator(java.util.Set concepts,
                        java.lang.String[] features)
```

## Methods

### getClassifierFeatureCombination

```
public java.util.Map getClassifierFeatureCombination(Concept concept)
```

CFL algorithm (Classifier Feature Learner) In this algorithm the best classifier with the best feature combination for a given concept is learned.

**Parameters:**

`concept` - The concept for which the cfc should be generated. If null, a cfc for all concepts will be returned.

**Returns:**

A map with the conceptID as key and the best classifier-feature combination for that concept as value.

### getClassifierFeatureCombination

```
public java.util.Map getClassifierFeatureCombination()
```

### main

```
public static void main(java.lang.String[] args)
```

(continued from last page)

For concept 17 classifier 1 with RMSE of 0.22196034583404162 and feature combination length sources entityTrust class has been found For concept 1 classifier 1 with RMSE of 0.21799234196876005 and feature combination sourceTrust class has been found For concept 18 classifier 3 with RMSE of 0.0 and feature combination length class has been found For concept 3 classifier 3 with RMSE of 0.22360679774997896 and feature combination length wordCount wordLength numericStart numericCount sources extractionTypes class has been found For concept 6 classifier 3 with RMSE of 0.31622776601683794 and feature combination wordLength numericCount sourceTrust entityTrust class has been found For concept 8 classifier 3 with RMSE of 0.0 and feature combination wordCount numericCount class has been found For concept 10 classifier 1 with RMSE of 0.2914830673506133 and feature combination wordCount class has been found For concept 12 classifier 3 with RMSE of 0.0 and feature combination entityTrust class has been found For concept 13 classifier 3 with RMSE of 0.0 and feature combination wordCount wordLength class has been found For concept 15 classifier 1 with RMSE of 0.1835495977760554 and feature combination wordCount wordLength numericStart numericEnd numericCount sourceTrust class has been found

# tud.iir.classification

## Class FeatureObject

java.lang.Object

└--tud.iir.classification.FeatureObject

```
public class FeatureObject
extends java.lang.Object
```

An object holding features.

**Author:**

David Urbansky, Philipp Katz

## Constructors

### FeatureObject

```
public FeatureObject(java.lang.Double[] features,
                     java.lang.String[] featureNames)
```

Create a feature object with a feature vector of doubles. The last index of the features must be 0 or 1 and refers to the class.

**Parameters:**

features - the features

featureNames - the feature names

### FeatureObject

```
public FeatureObject(java.util.Map features,
                     java.lang.Double classAssociation)
```

Instantiates a new feature object.

**Parameters:**

features - the features

classAssociation - the class association

### FeatureObject

```
public FeatureObject(java.util.Map features)
```

Instantiates a new feature object.

**Parameters:**

features - the features

## Methods

### getFeatures

```
public java.lang.Double[] getFeatures()
```

Gets the features.

(continued from last page)

**Returns:**  
the features

---

## setFeatures

```
public void setFeatures(java.lang.Double[] features)
```

Sets the features.

**Parameters:**  
features - the new features

---

## getFeatureNames

```
public java.lang.String[] getFeatureNames()
```

Gets the feature names.

**Returns:**  
the feature names

---

## setFeatureNames

```
public void setFeatureNames(java.lang.String[] featureNames)
```

Sets the feature names.

**Parameters:**  
featureNames - the featureNames as StringArray

---

## getClassAssociation

```
public int getClassAssociation()
```

---

## getClassAssociationAsString

```
public java.lang.String getClassAssociationAsString()
```

Gets the class association as string.

**Returns:**  
the class association as string

---

## setClassAssociation

```
public void setClassAssociation(int classAssociation)
```

Sets the class association.

**Parameters:**  
classAssociation - the new class association

---

## getFeature

```
public java.lang.Double getFeature(java.lang.String featureName)
```

---

(continued from last page)

Get a feature by its featureName.

**Parameters:**

featureName

**Returns:**

value of the specified featureName, or `null` if no feature with specified name, or no featureNames specified at all.

---

## toString

```
public java.lang.String toString()
```

# tud.iir.classification

## Class Helper

java.lang.Object

└--tud.iir.classification.Helper

---

public class **Helper**  
extends java.lang.Object

All methods of this class help importing and exporting data between the database and CSV files.

**Author:**

David Urbansky

---

## Constructors

### Helper

public **Helper**()

## Methods

### importEntityAssessmentData

public void **importEntityAssessmentData**()

Import hand chosen training and testing data for entity assessment. The file contains several concepts with classified entities for training and testing.

---

### main

public static void **main**(java.lang.String[] args)

**Parameters:**

args



# tud.iir.classification

## Class Stopwords

```
java.lang.Object
├-- java.util.AbstractCollection
│   ├── java.util.AbstractSet
│   │   ├── java.util.HashSet
│   │   └-- tud.iir.classification.Stopwords
```

### All Implemented Interfaces:

java.util.Collection, java.util.Set, java.io.Serializable, java.lang.Cloneable, java.util.Set

---

```
public class Stopwords
    extends java.util.HashSet
```

List of stopwords. Use the constants [STOP\\_WORDS\\_EN](#) or [STOP\\_WORDS\\_DE](#) for initialization with pre-defined stopword list. TODO when using Toolkit JAR in another project, the stopwords have to be copied to this project now. Use class.getResource() to avoid this? <http://www.devx.com/tips/Tip/5697>

### Author:

Philipp Katz

---

## Fields

### STOP\_WORDS\_EN

```
public static final java.lang.String STOP_WORDS_EN
```

Constant value: `config/stopwords_en.txt`

---

### STOP\_WORDS\_DE

```
public static final java.lang.String STOP_WORDS_DE
```

Constant value: `config/stopwords_de.txt`

---

## Constructors

### Stopwords

```
public Stopwords()
```

---

### Stopwords

```
public Stopwords(java.lang.String filePath)
```

---

## Methods

---

(continued from last page)

## addFromFile

```
public void addFromFile(java.lang.String filePath)
```

Add stopwords from file. One word each line, lines with # are treated as comments.

### Parameters:

filePath

---

## add

```
public boolean add(java.lang.String e)
```

---

## contains

```
public boolean contains(java.lang.Object o)
```

---

## toString

```
public java.lang.String toString()
```

---

## main

```
public static void main(java.lang.String[] args)
```

---

# tud.iir.classification

## Class Term

java.lang.Object

└─ tud.iir.classification.Term

All Implemented Interfaces:

java.io.Serializable

---

```
public class Term
    extends java.lang.Object
    implements java.io.Serializable
```

A term is a sequence of characters.

**Author:**

David Urbansky, Philipp Katz

---

## Constructors

### Term

```
public Term(java.lang.String text)
```

## Methods

### getText

```
public java.lang.String getText()
```

---

### lowerCaseText

```
public void lowerCaseText()
```

---

### equals

```
public boolean equals(java.lang.Object obj)
```

---

### hashCode

```
public int hashCode()
```

---

(continued from last page)

## toString

```
public java.lang.String toString()
```

---

## main

```
public static void main(java.lang.String[] a)
```

# tud.iir.classification

## Class WordCorrelation

java.lang.Object

└─ tud.iir.classification.WordCorrelation

All Implemented Interfaces:

java.io.Serializable

---

```
public class WordCorrelation
extends java.lang.Object
implements java.io.Serializable
```

Author:

David Urbansky, Klemens Muthmann, Sandro Reichert

---

## Constructors

### WordCorrelation

```
public WordCorrelation(Term word1,
                       Term word2)
```

---

## Methods

### setWordPair

```
public void setWordPair(Term word1,
                        Term word2)
```

---

### hashCode

```
public int hashCode()
```

---

### equals

```
public boolean equals(java.lang.Object obj)
```

---

### getTerm1

```
public Term getTerm1()
```

---

(continued from last page)

## getTerm2

```
public Term getTerm2()
```

---

## getAbsoluteCorrelation

```
public double getAbsoluteCorrelation()
```

---

## setAbsoluteCorrelation

```
public void setAbsoluteCorrelation(double absoluteCorrelation)
```

---

## increaseAbsoluteCorrelation

```
public void increaseAbsoluteCorrelation(double d)
```

---

## setRelativeCorrelation

```
public void setRelativeCorrelation(double relativeCorrelation)
```

---

## getRelativeCorrelation

```
public double getRelativeCorrelation()
```

---

## toString

```
public java.lang.String toString()
```

---

# tud.iir.classification

## Class WordCorrelationMatrix

```
java.lang.Object
|
|--tud.iir.classification.WordCorrelationMatrix
```

All Implemented Interfaces:  
java.io.Serializable

Direct Known Subclasses:  
[FastWordCorrelationMatrix](#)

```
public class WordCorrelationMatrix
extends java.lang.Object
implements java.io.Serializable
```

Correlation matrix.

See corresponding test case `WordCorrelationMatrixTest` for an example.

2010-08-04 -- changed internal data structure from HashSet to HashMap for performance optimizations. Serializations which have been created for the old class will not be compatible. Sorry.

**Author:**

David Urbansky, Klemens Muthmann, Sandro Reichert, Philipp Katz

## Constructors

### WordCorrelationMatrix

```
public WordCorrelationMatrix()
```

## Methods

### updatePair

```
public void updatePair(Term word1,
                       Term word2)
```

Add one to the correlation count of two terms. The order of the terms does not matter: t1,t2 = t2,t1

**Parameters:**

word1 - The first term.  
word2 - The second term.

### updatePair

```
public void updatePair(java.lang.String word1,
                       java.lang.String word2)
```

Add one to the correlation count of two terms. The order of the terms does not matter: t1,t2 = t2,t1

(continued from last page)

**Parameters:**

word1 - The first term.

word2 - The second term.

---

## makeRelativeScores

```
public void makeRelativeScores()
```

The co-occurrences are saved in the matrix as absolute values. They can be made relative by dividing through the total number of documents.

---

## getCorrelation

```
public WordCorrelation getCorrelation(Term word1,  
                                     Term word2)
```

---

## getCorrelation

```
public WordCorrelation getCorrelation(java.lang.String word1,  
                                     java.lang.String word2)
```

---

## getCorrelations

```
public java.util.List getCorrelations(java.lang.String word,  
                                     int minCooccurrences)
```

---

## getCorrelations

```
public java.util.Set getCorrelations()
```

Return all correlation pairs.

**Returns:**

---

## toString

```
public java.lang.String toString()
```



---

Package

**tud.iir.classification.controlledtagging**

# tud.iir.classification.controlledtagging

## Class ControlledTagger

java.lang.Object

└─tud.iir.classification.controlledtagging.ControlledTagger

---

public class **ControlledTagger**  
extends java.lang.Object

A TF-IDF and tag correlation based tagger using a controlled and weighted vocabulary. rem: enable assertions for debugging, VM arg -ea

**Author:**  
Philipp Katz

---

## Constructors

### ControlledTagger

public **ControlledTagger**()

Default constructor.

## Methods

### train

public void **train**(java.lang.String text,  
                  <any> tags)

---

### train

public void **train**(java.lang.String text)

Allows training, only with text. This can be used to build up an initial IDF index.

**Parameters:**

text

---

### train

public void **train**(<any> tags)

Allows training, only with tags. Each Bag of tags is considered as one training instance, e.g. one document. This builds up the tag vocabulary and the tag correlations.

**Parameters:**

tags

---

### addToVocabulary

public void **addToVocabulary**(java.lang.String tag)

---

(continued from last page)

---

## addToVocabulary

```
public int addToVocabulary(java.util.Collection tags)
```

---

## addToVocabularyFromFile

```
public int addToVocabularyFromFile(java.lang.String filePath)
```

**Deprecated.** *Use `train(String, Bag)` instead.*

Add controlled tagging vocabulary from text file. One line per tag + count. E.g.: design#26693 reference#25222 tools#24470 ... TODO use a different separator character, else we can not tag C# TODO remodel this for DeliciousCrawler files.

**Parameters:**

filePath

**Returns:**

---

## tag

```
public java.util.List tag(java.lang.String text)
```

Tag the supplied text.

**Parameters:**

text

**Returns:**

Array with assigned Tags, sorted by weight or empty List. Never `null`.

---

## tag

```
public java.util.Map tag(java.util.Collection texts)
```

---

## normalize

```
public <any> normalize(<any> tags)
```

Normalize a list of Tags according to the stemming rules. We need this for the evaluation process, as the the test Tags need to be normalized the same way.

**Parameters:**

tags

---

## getSettings

```
public ControlledTaggerSettings getSettings()
```

(continued from last page)

Set the fast mode. This is only relevant when using correlations, [ControlledTaggerSettings.TaggingCorrelationType.SHALLOW\\_CORRELATIONS](#) or [ControlledTaggerSettings.TaggingCorrelationType.DEEP\\_CORRELATIONS](#). When enabled, the potentially huge list of tag candidates will be pruned, before checking their correlations, which is computationally very expensive. The pruned list should still be big enough to account for re-rankings through correlations; I checked this extensively and set the pruned list size very conservatively. Anyway, to be absolutely sure, not to lose any accuracy, we can disable the pruning here. In my experiments, the tagging process took about 20 times longer without this optimization.

**Parameters:**`fastMode`

---

## setSettings

```
public void setSettings(ControlledTaggerSettings settings)
```

---

## getIndex

```
public ControlledTaggerIndex getIndex()
```

---

## save

```
public void save(java.lang.String filePath)
```

Serialize this tagger to disk.

**Parameters:**`filePath`

---

## load

```
public void load(java.lang.String filePath)
```

Load safed tagger index [ControlledTaggerIndex](#) from disk.

---

## toString

```
public java.lang.String toString()
```

Hook for the deserialization.

**Parameters:**`in`**Throws:**`IOException``ClassNotFoundException`

---

## writeDataToReport

```
public void writeDataToReport()
```

Write some statistical information concerning the index.

---

(continued from last page)

## main

```
public static void main(java.lang.String[] args)
    throws java.lang.Exception
```

# tud.iir.classification.controlledtagging

## Class ControlledTaggerEvaluation

```
java.lang.Object
├── tud.iir.classification.controlledtagging.DeliciousDatasetReader
│   ├── tud.iir.classification.controlledtagging.DeliciousDatasetSplitter
│   │   └── tud.iir.classification.controlledtagging.ControlledTaggerEvaluation
```

```
public class ControlledTaggerEvaluation
extends DeliciousDatasetSplitter
```

Evaluator for the [ControlledTagger](#) using the delicious data set T140. Important: VM args -Xmx1024M

**Author:**  
Philipp Katz

## Constructors

### ControlledTaggerEvaluation

```
public ControlledTaggerEvaluation()
```

## Methods

### evaluate

```
public ControlledTaggerEvaluationResult evaluate(ControlledTaggerEvaluationSettings settings)
```

Evaluate with the specified [ControlledTaggerEvaluationSettings](#).

### evaluate

```
public void evaluate(java.util.List settings,
    java.lang.String resultFilePath)
```

Do evaluation with a list of different settings, save result to textfile. The result file contains one line for each evaluation step with settings and corresponding results.

**Parameters:**

```
settings
resultFilePath
```

### train

```
public void train(DeliciousDatasetReader.DatasetEntry entry,
    int index)
```

(continued from last page)

## test

```
public void test(DeliciousDatasetReader.DatasetEntry entry,  
                int index)
```

---

## startTrain

```
public void startTrain()
```

---

## finishTrain

```
public void finishTrain()
```

---

## startTest

```
public void startTest()
```

---

## finishTest

```
public void finishTest()
```

---

## getEvaluationResult

```
public ControlledTaggerEvaluationResult getEvaluationResult()
```

---

## main

```
public static void main(java.lang.String[] args)
```

---

# tud.iir.classification.controlledtagging Class ControlledTaggerEvaluationResult

java.lang.Object

└-tud.iir.classification.controlledtagging.ControlledTaggerEvaluationResult

public class **ControlledTaggerEvaluationResult**  
extends java.lang.Object

Keeps results concerning the Tagger evaluation for specific [ControlledTaggerEvaluationSettings](#) like Pr/Rc/F1, etc.

**Author:**  
Philipp Katz

## Constructors

### ControlledTaggerEvaluationResult

public **ControlledTaggerEvaluationResult**()

## Methods

### addTestResult

public void **addTestResult**(double precision,  
double recall,  
int assignedTags)

### getAvgPrecision

public double **getAvgPrecision**()

### getAvgRecall

public double **getAvgRecall**()

### getAvgFOne

public double **getAvgFOne**()

### getAvgTagCount

public double **getAvgTagCount**()



(continued from last page)

---

## getTaggedEntryCount

```
public int getTaggedEntryCount()
```

---

## startTraining

```
public void startTraining()
```

---

## stopTraining

```
public void stopTraining()
```

---

## startTesting

```
public void startTesting()
```

---

## stopTesting

```
public void stopTesting()
```

---

## getTestStop

```
public StopWatch getTestStop()
```

---

## getTrainStop

```
public StopWatch getTrainStop()
```

---

## toString

```
public java.lang.String toString()
```

---

## printStatistics

```
public void printStatistics()
```

# tud.iir.classification.controlledtagging

## Class ControlledTaggerEvaluationSettings

java.lang.Object

```

+--tud.iir.classification.controlledtagging.ControlledTaggerSettings
    +--tud.iir.classification.controlledtagging.ControlledTaggerEvaluationSettings
  
```

public class **ControlledTaggerEvaluationSettings**  
 extends [ControlledTaggerSettings](#)

Extends [ControlledTaggerEvaluationSettings](#) with evaluation specific parameters.

**Author:**  
 Philipp Katz

## Constructors

### ControlledTaggerEvaluationSettings

```
public ControlledTaggerEvaluationSettings()
```

### ControlledTaggerEvaluationSettings

```

public ControlledTaggerEvaluationSettings(int trainLimit,
                                         int testLimit,
                                         ControlledTaggerSettings.TaggingType
taggingType,
ControlledTaggerSettings.TaggingCorrelationType correlationType,
                                         float tfidfThreshold,
                                         int tagCount,
                                         float correlationWeight,
                                         float priorWeight)
  
```

Monstous nearly-all-parameter-constructor for evaluation.

#### Parameters:

```

taggingType
correlationType
tfidfThreshold
tagCount
correlationWeight
priorWeight
tagMatchPattern
stopwords
trainLimit
testLimit
  
```

## Methods

### getTrainLimit

```
public int getTrainLimit()
```

(continued from last page)

**Returns:**  
the trainLimit

---

## setTrainLimit

```
public void setTrainLimit(int trainLimit)
```

**Parameters:**  
trainLimit - the trainLimit to set

---

## getTestLimit

```
public int getTestLimit()
```

**Returns:**  
the testLimit

---

## setTestLimit

```
public void setTestLimit(int testLimit)
```

**Parameters:**  
testLimit - the testLimit to set

---

## toString

```
public java.lang.String toString()
```

# tud.iir.classification.controlledtagging

## Class ControlledTaggerIndex

java.lang.Object

└--tud.iir.classification.controlledtagging.ControlledTaggerIndex

All Implemented Interfaces:

java.io.Serializable

---

```
public class ControlledTaggerIndex
  extends java.lang.Object
  implements java.io.Serializable
```

The ControlledTaggerIndex contains all necessary index data for the Tagger. This includes the controlled vocabulary, word correlations, stems. This class can be serialized to disk via the Tagger.

**Author:**

Philipp Katz

---

## Methods

### getIdfIndex

```
public <any> getIdfIndex()
```

---

### setIdfIndex

```
public void setIdfIndex(<any> idfIndex)
```

---

### getTagVocabulary

```
public <any> getTagVocabulary()
```

---

### setTagVocabulary

```
public void setTagVocabulary(<any> tagVocabulary)
```

---

### getStemmedTagVocabulary

```
public <any> getStemmedTagVocabulary()
```

---

### setStemmedTagVocabulary

```
public void setStemmedTagVocabulary(<any> stemmedTagVocabulary)
```

(continued from last page)

---

## getUnstemMap

```
public java.util.Map getUnstemMap()
```

---

---

## setUnstemMap

```
public void setUnstemMap(java.util.Map unstemMap)
```

---

---

## getIdfCount

```
public int getIdfCount()
```

---

---

## setIdfCount

```
public void setIdfCount(int idfCount)
```

---

---

## getTrainCount

```
public int getTrainCount()
```

---

---

## setTrainCount

```
public void setTrainCount(int trainCount)
```

---

---

## getAverageTagOccurence

```
public float getAverageTagOccurence()
```

---

---

## setAverageTagOccurence

```
public void setAverageTagOccurence(float averageTagOccurence)
```

---

---

## getWcm

```
public WordCorrelationMatrix getWcm()
```

---

(continued from last page)

## setWcm

```
public void setWcm(WordCorrelationMatrix wcm)
```

---

## isDirtyIndex

```
public boolean isDirtyIndex()
```

---

## setDirtyIndex

```
public void setDirtyIndex(boolean dirtyIndex)
```

---

## toString

```
public java.lang.String toString()
```

---

# tud.iir.classification.controlledtagging Class ControlledTaggerSettings

java.lang.Object

└-tud.iir.classification.controlledtagging.ControlledTaggerSettings

Direct Known Subclasses:

[ControlledTaggerEvaluationSettings](#)

```
public class ControlledTaggerSettings
extends java.lang.Object
```

This class bundles all settings for the [ControlledTagger](#).

Author:

Philipp Katz

## Fields

### DEFAULT\_TFIDF\_THRESHOLD

```
public static final float DEFAULT_TFIDF_THRESHOLD
```

Constant value: 0.0050

### DEFAULT\_TAG\_COUNT

```
public static final int DEFAULT_TAG_COUNT
```

Constant value: 10

### DEFAULT\_CORRELATION\_WEIGHT

```
public static final float DEFAULT_CORRELATION_WEIGHT
```

Constant value: 50.0

### DEFAULT\_PRIOR\_WEIGHT

```
public static final float DEFAULT_PRIOR_WEIGHT
```

Constant value: 1.0

### DEFAULT\_TAG\_MATCH\_PATTERN

```
public static final java.util.regex.Pattern DEFAULT_TAG_MATCH_PATTERN
```

## Constructors

(continued from last page)

## ControlledTaggerSettings

```
public ControlledTaggerSettings(ControlledTaggerSettings.TaggingType taggingType,  
                                ControlledTaggerSettings.TaggingCorrelationType  
                                correlationType,  
                                float tfidfThreshold,  
                                int tagCount,  
                                float correlationWeight,  
                                float priorWeight,  
                                java.util.regex.Pattern tagMatchPattern,  
                                java.util.Set stopwords)
```

---

## ControlledTaggerSettings

```
public ControlledTaggerSettings()
```

## Methods

### getTaggingType

```
public ControlledTaggerSettings.TaggingType getTaggingType()
```

---

### setTaggingType

```
public void setTaggingType(ControlledTaggerSettings.TaggingType taggingType)
```

---

### getCorrelationType

```
public ControlledTaggerSettings.TaggingCorrelationType getCorrelationType()
```

---

### setCorrelationType

```
public void setCorrelationType(ControlledTaggerSettings.TaggingCorrelationType  
correlationType)
```

---

### getTfidfThreshold

```
public float getTfidfThreshold()
```

---

### setTfidfThreshold

```
public void setTfidfThreshold(float tfidfThreshold)
```

Set the threshold for the TFIDF value when in [ControlledTaggerSettings.TaggingType.THRESHOLD](#) mode.



(continued from last page)

**Parameters:**tfidfThreshold

---

**getTagCount**

```
public int getTagCount()
```

---

**setTagCount**

```
public void setTagCount(int tagCount)
```

Set max. number of tags to assign when in [ControlledTaggerSettings.TaggingType.FIXED\\_COUNT](#) mode.

**Parameters:**tagCount

---

**getCorrelationWeight**

```
public float getCorrelationWeight()
```

---

**setCorrelationWeight**

```
public void setCorrelationWeight(float correlationWeight)
```

---

**getPriorWeight**

```
public float getPriorWeight()
```

---

**setPriorWeight**

```
public void setPriorWeight(float priorWeight)
```

When enabled, tags from the controlled vocabulary which have a high occurrence are preferred. Set to -1 to disable.

**Parameters:**usePriors

---

**getTagMatchPattern**

```
public java.util.regex.Pattern getTagMatchPattern()
```

---

**setTagMatchPattern**

```
public void setTagMatchPattern(java.util.regex.Pattern tagMatchPattern)
```

---

---

## getStopwords

```
public java.util.Set getStopwords()
```

---

## setStopwords

```
public void setStopwords(java.util.Set stopwords)
```

Set the Set of Stopwords to use, for example [Stopwords](#).

### Parameters:

stopwords

---

## getStemmer

```
public SnowballStemmer getStemmer()
```

---

## setStemmer

```
public void setStemmer(SnowballStemmer stemmer)
```

---

## toString

```
public java.lang.String toString()
```

---

## main

```
public static void main(java.lang.String[] args)
```

# tud.iir.classification.controlledtagging Class ControlledTaggerSettings.TaggingType

java.lang.Object

└─ java.lang.Enum

└─ tud.iir.classification.controlledtagging.ControlledTaggerSettings.TaggingType

All Implemented Interfaces:

java.io.Serializable, java.lang.Comparable

---

public static final class **ControlledTaggerSettings.TaggingType**  
extends java.lang.Enum

---

## Fields

### THRESHOLD

```
public static final  
tud.iir.classification.controlledtagging.ControlledTaggerSettings.TaggingType  
THRESHOLD
```

---

### FIXED\_COUNT

```
public static final  
tud.iir.classification.controlledtagging.ControlledTaggerSettings.TaggingType  
FIXED_COUNT
```

---

## Methods

### values

```
public static ControlledTaggerSettings.TaggingType\[\] values()
```

---

### valueOf

```
public static ControlledTaggerSettings.TaggingType valueOf(java.lang.String name)
```

# tud.iir.classification.controlledtagging Class ControlledTaggerSettings.TaggingCorrelationType

```
java.lang.Object
  |
  +- java.lang.Enum
        |
        +-
```

```
tud.iir.classification.controlledtagging.ControlledTaggerSettings.TaggingCorrelationType
```

All Implemented Interfaces:

java.io.Serializable, java.lang.Comparable

```
public static final class ControlledTaggerSettings.TaggingCorrelationType
extends java.lang.Enum
```

## Fields

### NO\_CORRELATIONS

```
public static final
tud.iir.classification.controlledtagging.ControlledTaggerSettings.TaggingCorrelationType
NO_CORRELATIONS
```

### SHALLOW\_CORRELATIONS

```
public static final
tud.iir.classification.controlledtagging.ControlledTaggerSettings.TaggingCorrelationType
SHALLOW_CORRELATIONS
```

### DEEP\_CORRELATIONS

```
public static final
tud.iir.classification.controlledtagging.ControlledTaggerSettings.TaggingCorrelationType
DEEP_CORRELATIONS
```

## Methods

### values

```
public static ControlledTaggerSettings.TaggingCorrelationType\[\] values()
```

### valueOf

```
public static ControlledTaggerSettings.TaggingCorrelationType valueOf(java.lang.String
name)
```

(continued from last page)

# tud.iir.classification.controlledtagging Class DeliciousDatasetReader

java.lang.Object

└─tud.iir.classification.controlledtagging.DeliciousDatasetReader

Direct Known Subclasses:

[DeliciousDatasetSplitter](#)

```
public class DeliciousDatasetReader
extends java.lang.Object
```

Parser for Delicious data set from <http://nlp.uned.es/social-tagging/delicioust140/> See main method for usage example.

Author:

Philipp Katz

## Constructors

### DeliciousDatasetReader

```
public DeliciousDatasetReader()
```

## Methods

### read

```
public void read(DeliciousDatasetReader.DatasetCallback callback)
```

Start reading the dataset, using the specified callback.

Parameters:

callback

### read

```
public void read(DeliciousDatasetReader.DatasetCallback callback,
                int limit)
```

Start reading the dataset, using the specified callback.

Parameters:

callback

limit - the number of entries to read, or -1 for no limit.

### read

```
public void read(DeliciousDatasetReader.DatasetCallback callback,
                int limit,
                int offset)
```

Start reading the dataset, using the specified callback.

(continued from last page)

**Parameters:**

callback

limit - the number of entries to read, or -1 for no limit.

offset - the offset where to start reading, or 0 for no offset.

---

## setDataPath

```
public void setDataPath(java.lang.String dataPath)
```

Set the path to the data files. One can obtain them from <http://nlp.uned.es/social-tagging/delicioust140/> -- download both ZIP files and put their contents "taginfo.xml" and "fddocuments" in one directory.

**Parameters:**

dataPath

---

## setFilter

```
public void setFilter(DeliciousDatasetReader.DatasetFilter filter)
```

Set filter for entries.

**Parameters:**

filter

---

## main

```
public static void main(java.lang.String[] args)
```

## tud.iir.classification.controlledtagging Class DeliciousDatasetReader.DatasetCallback

java.lang.Object

└-tud.iir.classification.controlledtagging.DeliciousDatasetReader.DatasetCallback

---

public static abstract class **DeliciousDatasetReader.DatasetCallback**  
extends java.lang.Object

---

### Constructors

#### DeliciousDatasetReader.DatasetCallback

public **DeliciousDatasetReader.DatasetCallback**()

### Methods

#### callback

public abstract void **callback**([DeliciousDatasetReader.DatasetEntry](#) entry)

---

#### stop

public final void **stop**()



# tud.iir.classification.controlledtagging

## Class DeliciousDatasetReader.DatasetEntry

java.lang.Object

└--tud.iir.classification.controlledtagging.DeliciousDatasetReader.DatasetEntry

---

public class **DeliciousDatasetReader.DatasetEntry**  
extends java.lang.Object

Represents an entry in the data set. TODO use Tag class instead of Bag, also change return type.

**Author:**

Philipp Katz

---

## Constructors

### DeliciousDatasetReader.DatasetEntry

public **DeliciousDatasetReader.DatasetEntry**()

## Methods

### toString

public java.lang.String **toString**()

---

### getUrl

public java.lang.String **getUrl**()

get entry's url.

---

### getFiletype

public java.lang.String **getFiletype**()

get file type of associated file.

---

### getNumUsers

public int **getNumUsers**()

get the number of users who bookmarked this entry.

---

### getPath

public java.lang.String **getPath**()

get the path to the associated file.

---

---

## getFile

```
public java.io.File getFile()
```

get associated file.

---

## getTags

```
public <any> getTags()
```

---

## getAssignedTags

```
public java.util.List getAssignedTags()
```

# tud.iir.classification.controlledtagging

## Class DeliciousDatasetReader.DatasetFilter

java.lang.Object

└--tud.iir.classification.controlledtagging.DeliciousDatasetReader.DatasetFilter

---

public static class **DeliciousDatasetReader.DatasetFilter**  
extends java.lang.Object

Allows to filter DatasetEntries based on their attributes. Available Filetypes are html, pdf, xml or swf.

**Author:**

Philipp Katz

---

## Constructors

### DeliciousDatasetReader.DatasetFilter

public **DeliciousDatasetReader.DatasetFilter**()

## Methods

### setAllowedFiletypes

public void **setAllowedFiletypes**(java.util.Collection allowedFiletypes)

---

### addAllowedFiletype

public void **addAllowedFiletype**(java.lang.String allowedFiletype)

---

### setMinUsers

public void **setMinUsers**(int minUsers)

---

### setMinUserTagRatio

public void **setMinUserTagRatio**(double minUserTagRatio)

---

### setMaxFileSize

public void **setMaxFileSize**(int maxFileSize)

Limit for maximum accepted file size in bytes. This is useful, because very big HTML files can cause the HTML parser to stall. I usually set this to 600.000, to skip files above 600 kB. Set to -1 for no limit.

(continued from last page)

**Parameters:**

maxFileSize

# tud.iir.classification.controlledtagging Class DeliciousDatasetSplitter

java.lang.Object

├- [tud.iir.classification.controlledtagging.DeliciousDatasetReader](#)  
└- **tud.iir.classification.controlledtagging.DeliciousDatasetSplitter**

Direct Known Subclasses:

[ControlledTaggerEvaluation](#)

---

public abstract class **DeliciousDatasetSplitter**  
extends [DeliciousDatasetReader](#)

Extends DeliciousDatasetReader with random splitting capabilities for evaluation purposes. For now, we use a fixed split 50:50.

**Author:**

Philipp Katz

---

## Constructors

### DeliciousDatasetSplitter

public **DeliciousDatasetSplitter**()

## Methods

### calculateSplit

public void **calculateSplit**()

The split is calculated upon initialization. Call this, to calculate a new split.

---

### read

public void **read**()

---

### readTest

public void **readTest**()

---

### readTrain

public void **readTrain**()

---

(continued from last page)

## train

```
public abstract void train(DeliciousDatasetReader.DatasetEntry entry,  
    int index)
```

---

## test

```
public abstract void test(DeliciousDatasetReader.DatasetEntry entry,  
    int index)
```

---

## startTrain

```
public void startTrain()
```

---

## finishTrain

```
public void finishTrain()
```

---

## startTest

```
public void startTest()
```

---

## finishTest

```
public void finishTest()
```

---

## setTrainLimit

```
public void setTrainLimit(int trainLimit)
```

---

## setTestLimit

```
public void setTestLimit(int testLimit)
```

---

# tud.iir.classification.controlledtagging

## Class Tag

java.lang.Object

└─tud.iir.classification.controlledtagging.Tag

---

public class **Tag**  
extends java.lang.Object

Represents a Tag.

**Author:**  
Philipp Katz

---

## Constructors

### Tag

public **Tag**()

---

### Tag

public **Tag**(java.lang.String name,  
float weight)

---

## Methods

### getName

public java.lang.String **getName**()

---

### setName

public void **setName**(java.lang.String name)

---

### getWeight

public float **getWeight**()

The Tag's weight, which is based on tf-idf, but which may be altered by various re-ranking processes.

**Returns:**

---

(continued from last page)

## setWeight

```
public void setWeight(float weight)
```

---

## increaseWeight

```
public void increaseWeight(float by)
```

---

## getOriginalWeight

```
public float getOriginalWeight()
```

The Tag's weight, determined by tf-idf. Immutable. TODO I dont like this solution. Think this over again, remove this method/field to weight and the other one to "internal"/"ranking" weight etc., make it accessible just for the tagger, as it makes no sense to expose this to the outside.

**Returns:**

---

## toString

```
public java.lang.String toString()
```

---

## hashCode

```
public int hashCode()
```

---

## equals

```
public boolean equals(java.lang.Object obj)
```



# tud.iir.classification.controlledtagging

## Class TagComparator

java.lang.Object

└--tud.iir.classification.controlledtagging.TagComparator

### All Implemented Interfaces:

java.util.Comparator

---

```
public class TagComparator
    extends java.lang.Object
    implements java.util.Comparator
```

Compare Tags based on their weights.

### Author:

Philipp Katz

---

## Constructors

### TagComparator

```
public TagComparator()
```

Create new descending TagComparator.

---

### TagComparator

```
public TagComparator(boolean descending)
```

Create new TagComparator.

### Parameters:

*descending* - if true, Tags are sorted descendingly by their weights, false ascendingly.

---

## Methods

### compare

```
public int compare(Tag t1,
                  Tag t2)
```

---

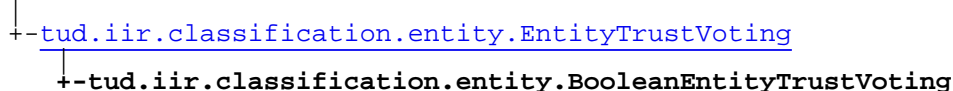
Package

**tud.iir.classification.entity**

# tud.iir.classification.entity

## Class BooleanEntityTrustVoting

java.lang.Object



All Implemented Interfaces:

[EntityTrustVotingInterface](#)

```

public class BooleanEntityTrustVoting
extends EntityTrustVoting
implements EntityTrustVotingInterface
  
```

## Constructors

### BooleanEntityTrustVoting

```
public BooleanEntityTrustVoting()
```

## Methods

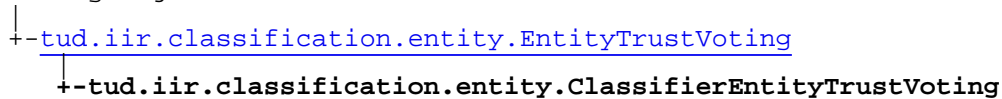
### runVoting

```
public void runVoting()
```

Boolean trust voting. for concept Movie entity "The Incredibles" has been found for concept Mobile Phone entity "Nokia" has been found for concept Notebook entity "Acer" has been found for concept Car entity "Audi" has been found for concept Song entity "Close To You" has been found for concept City entity "Boston" has been found for concept Country entity "India" has been found for concept Sport entity "Golf" has been found for concept Actor entity "Tommy Lee Jones" has been found run voting... 79956 entities\_sources were affected by page voting 616 sources were affected by entity voting run voting... 199588 entities\_sources were affected by page voting 548 sources were affected by entity voting run voting... 48927 entities\_sources were affected by page voting 29 sources were affected by entity voting run voting... 3848 entities\_sources were affected by page voting 2 sources were affected by entity voting run voting... 30 entities\_sources were affected by page voting 1 sources were affected by entity voting run voting... 26 entities\_sources were affected by page voting 0 sources were affected by entity voting run voting... 0 entities\_sources were affected by page voting 0 sources were affected by entity voting 1503 of 1910 sources have entity trust 1 332375 of 427360 entities have entity trust 1 stopped, runtime: 4140 seconds

# tud.iir.classification.entity Class ClassifierEntityTrustVoting

java.lang.Object



All Implemented Interfaces:

[EntityTrustVotingInterface](#)

public class **ClassifierEntityTrustVoting**  
extends [EntityTrustVoting](#)  
implements [EntityTrustVotingInterface](#)

## Constructors

### ClassifierEntityTrustVoting

public **ClassifierEntityTrustVoting**()

## Methods

### runVoting

public void **runVoting**()

Start classification using the best performing classifiers and feature combinations for each concept. ::: runtime: 27424.0 seconds, 04/04/2009

### runVoting

public void **runVoting**(int classifierType)

### main

public static void **main**(java.lang.String[] args)

# tud.iir.classification.entity

## Class EntityAssessor

java.lang.Object

└-tud.iir.classification.entity.EntityAssessor

Direct Known Subclasses:

[PMI](#), [RandomGraphWalk](#), [WordFeatureClassifier](#)

---

public abstract class **EntityAssessor**  
extends java.lang.Object

---

## Constructors

### EntityAssessor

public **EntityAssessor**()

---

## Methods

### logMetrics

public java.util.ArrayList **logMetrics**(java.util.HashSet concepts,  
java.util.HashMap evaluationMetrics)

# tud.iir.classification.entity Class EntityClassifier

java.lang.Object

└-[tud.iir.classification.Classifier](#)

└-tud.iir.classification.entity.EntityClassifier

---

public class **EntityClassifier**  
extends [Classifier](#)

---

## Constructors

### EntityClassifier

public **EntityClassifier**(int type)

## Methods

### trainClassifier

```
public boolean trainClassifier(int conceptID,  
                               java.sql.PreparedStatement featureString)
```

Train a classifier with the samples save in the database. The classifier is trained on a concept level.

**Parameters:**

conceptID - The id of the concept for which the classifier should be trained.

featureString - The SQL query string with the desired features to train the classifier.

---

### trainClassifier

```
public boolean trainClassifier(int conceptID,  
                               java.sql.PreparedStatement featureString,  
                               java.sql.PreparedStatement classificationString)
```

# tud.iir.classification.entity

## Class EntityTrustVoting

java.lang.Object

└--tud.iir.classification.entity.EntityTrustVoting

Direct Known Subclasses:

[BooleanEntityTrustVoting](#), [ClassifierEntityTrustVoting](#), [FeatureEntityTrustVoting](#), [GradualEntityTrustVoting](#)

```
public class EntityTrustVoting
extends java.lang.Object
```

## Constructors

### EntityTrustVoting

```
public EntityTrustVoting()
```

## Methods

### createEntityFile

```
public void createEntityFile()
```

Create an entity file. file format: concept: entity total | 1:x,2:y,....

### createEntityFile2

```
public void createEntityFile2()
```

file format: concept queryType entity

### createEntityTrustChart

```
public void createEntityTrustChart()
```

### findEntityConnection

```
public boolean findEntityConnection(int entityID1,
    int entityID2,
    int lastSourceID,
    java.util.HashSet usedSources,
    java.util.ArrayList pathArray)
```

find connection (sources-entities) between two entities (depth first)

(continued from last page)

## main

```
public static void main(java.lang.String[] args)
```

### Parameters:

args



## tud.iir.classification.entity Interface EntityTrustVotingInterface

All Known Implementing Classes:

[BooleanEntityTrustVoting](#), [ClassifierEntityTrustVoting](#), [FeatureEntityTrustVoting](#),  
[GradualEntityTrustVoting](#)

---

public interface **EntityTrustVotingInterface**  
extends

---

### Methods

#### runVoting

```
public void runVoting()
```

# tud.iir.classification.entity

## Class EvaluationHelper

```
java.lang.Object
└--tud.iir.classification.entity.EvaluationHelper
```

---

```
public class EvaluationHelper
extends java.lang.Object
```

The EvaluationHelper supports functions to create an evaluation set for entity assessment.

**Author:**  
David

---

## Constructors

### EvaluationHelper

```
public EvaluationHelper()
```

## Methods

### extract

```
public void extract()
```

Extract entities for given concept in the ontology. Also extract Search engine hit counts to estimate popularity.

---

### retrieveHitCounts

```
public void retrieveHitCounts()
```

retrieve PMI scores for evaluation entities popularity: hit count of entity alone popularity2: hit count of entity + concept of entity isX: hit count of query "ENTITY is a CONCEPT" XsuchAs: hit count of query "CONCEPTs such as ENTITY" XLike: hit count of query "CONCEPTs like ENTITY" Xincluding: hit count of query "CONCEPTs including ENTITY" AndOtherX: hit count of query "ENTITY and other CONCEPTs"

---

### main

```
public static void main(java.lang.String[] args)
```

#### Parameters:

args

---

# tud.iir.classification.entity Class FeatureEntityTrustVoting

java.lang.Object

└- tud.iir.classification.entity.EntityTrustVoting  
└- tud.iir.classification.entity.FeatureEntityTrustVoting

All Implemented Interfaces:

[EntityTrustVotingInterface](#)

public class **FeatureEntityTrustVoting**  
extends [EntityTrustVoting](#)  
implements [EntityTrustVotingInterface](#)

## Constructors

### FeatureEntityTrustVoting

public **FeatureEntityTrustVoting**()

## Methods

### runVoting

public void **runVoting**()

last run with assignSourceTrust("2"); on 24/03/2009 source trust has been assigned in 1952.0 seconds entity trust has been assigned in 273.0 seconds ::: runtime: 2226.0 seconds

### main

public static void **main**(java.lang.String[] args)

# tud.iir.classification.entity

## Class GradualEntityTrustVoting

java.lang.Object

```

+-- tud.iir.classification.entity.EntityTrustVoting
    +-- tud.iir.classification.entity.GradualEntityTrustVoting
  
```

All Implemented Interfaces:

[EntityTrustVotingInterface](#)

```

public class GradualEntityTrustVoting
extends EntityTrustVoting
implements EntityTrustVotingInterface
  
```

## Constructors

### GradualEntityTrustVoting

```
public GradualEntityTrustVoting()
```

## Methods

### runVoting

```
public void runVoting()
```

UPDATE sources SET voting = 0,entityTrust = 0; UPDATE entities SET voting = 0,trust = 0,class = null; ----- for concept Movie entity "Snow Dogs" has been found for concept Mobile Phone entity "Nokia" has been found for concept Notebook entity "Acer" has been found for concept Car entity "Audi" has been found for concept City entity "Boston" has been found for concept Song entity "Close To You" has been found for concept Country entity "India" has been found for concept Actor entity "Tommy Lee Jones" has been found for concept Sport entity "Golf" has been found run voting... 242 entities\_sources were affected by page voting 79956 sources were affected by entity voting run voting... 671 entities\_sources were affected by page voting 199432 sources were affected by entity voting run voting... 553 entities\_sources were affected by page voting 49030 sources were affected by entity voting run voting... 34 entities\_sources were affected by page voting 3901 sources were affected by entity voting run voting... 2 entities\_sources were affected by page voting 30 sources were affected by entity voting run voting... 1 entities\_sources were affected by page voting 26 sources were affected by entity voting run voting... 0 entities\_sources were affected by page voting 0 sources were affected by entity voting 0 of 24021 sources have entity trust 1 0 of 427360 entities have entity trust 1 stopped, runtime: 5304 seconds

### main

```
public static void main(java.lang.String[] args)
```

(continued from last page)

for concept Movie entity "Snow Dogs" has been found for concept Mobile Phone entity "Nokia" has been found for concept Notebook entity "Acer" has been found for concept Car entity "Audi" has been found for concept City entity "Boston" has been found for concept Song entity "Close To You" has been found for concept Country entity "India" has been found for concept Actor entity "Tommy Lee Jones" has been found for concept Sport entity "Golf" has been found run voting... 242 sources voted 79947 entities voted run voting... 671 sources voted 199432 entities voted run voting... 553 sources voted 49030 entities voted run voting... 34 sources voted 3901 entities voted run voting... 2 sources voted 30 entities voted run voting... 1 sources voted 26 entities voted run voting... 0 sources voted 0 entities voted final source voting finished stopped, runtime: 3937 seconds ##### after removing indices on voting and trust: 1565 seconds

## tud.iir.classification.entity

### Class NoisyOr

java.lang.Object

└--tud.iir.classification.entity.NoisyOr

---

```
public class NoisyOr
    extends java.lang.Object
```

The Noisy-Or formula for assessment of (un)supervised information extraction. Noisy-Or as described in "A Probabilistic Model of Redundancy in Information Extraction, 2006".

**Author:**

David

---

## Constructors

### NoisyOr

```
public NoisyOr()
```

## Methods

### classify

```
public boolean classify(Entity entity)
```

---

### main

```
public static void main(java.lang.String[] args)
```

**Parameters:**

args

# tud.iir.classification.entity

## Class PMI

```
java.lang.Object
├── tud.iir.classification.entity.EntityAssessor
│   └── tud.iir.classification.entity.PMI
```

```
public class PMI
extends EntityAssessor
```

Implementation similar to the one described in the KnowItAll system:

- Information Extraction from the Web: Techniques and Applications, Alexander Yates, 2007, page 43
- The Use of Web-based Statistics to Validate Information Extraction, Stephen Soderland, Oren Etzioni, Tal Shaked, and Daniel S. Weld, 2004
- WebScale Information Extraction in KnowItAll (Preliminary Results), Etzioni et al., 2004

Difference: No bootstrapping to find discriminators for each class, but use generic ones. Workflow: 1. Find instances using discriminators. 2. Calculate prior probabilities  $P(I)$  and  $P(-I)$  by manually checking the extractions.  $I$  = correct instance. 3. Take  $k$  ( $k \sim 10$ ) positive and  $k$  negative instances (negative are positives from another class). 4. Calculate PMIs for all discriminators and all seeds of the training set. 5. Find a threshold that splits positive and negative instances. 6. Create a tuning set of another  $k$  positive and  $k$  negative instances. 7. Calculate for all discriminators and instances  $P(\text{PMI} > \text{thresh} \mid \text{class})$ ,  $P(\text{PMI} > \text{thresh} \mid -\text{class})$ ,  $P(\text{PMI} \leq \text{thresh} \mid \text{class})$ ,  $P(\text{PMI} \leq \text{thresh} \mid -\text{class})$  by simply counting the correct/incorrect classifications. 8. Use trained probabilities  $P(\text{PMI}(I,D) > \text{thresh} \mid \text{class}) = P(f_i|I)$  in NBC. 9. New instances can now be classified using NBC.

Author:  
David

## Constructors

### PMI

```
public PMI()
```

## Methods

### addConcept

```
public void addConcept(Concept c)
```

### classifySoft

```
public java.lang.Double[] classifySoft(Entity entity)
```

---

## classify

```
public boolean classify(Entity entity)
```

---

## evaluate

```
public void evaluate()
```

Evaluate the algorithm by classifying entities with a score above the threshold as true and calculating precision and recall using the test entities.

---

## main

```
public static void main(java.lang.String[] args)
```



# tud.iir.classification.entity

## Class RandomGraphWalk

```
java.lang.Object
├── tud.iir.classification.entity.EntityAssessor
│   └── tud.iir.classification.entity.RandomGraphWalk
```

```
public class RandomGraphWalk
extends EntityAssessor
```

The Random Graph Walk assesses supervised information extraction. Algorithm similar to the one explained in

- Language Independent Set Expansion of Named Entities Using the Web, 2007
- Automatic Set Instance Extraction using the Web, 2009
- Iterative Set Expansion of Named Entities using the Web, 2008

**Author:**

David Urbansky

## Constructors

### RandomGraphWalk

```
public RandomGraphWalk()
```

## Methods

### evaluate

```
public void evaluate()
```

Evaluate the algorithm by classifying entities with a score above the threshold as true and calculating precision and recall using the test entities.

### matrixTest

```
public void matrixTest()
```

Example graph walk:

E:\Projects\Programming\Java\WebKnox\documentationImages\graphWalkExample.png

### classify

```
public boolean classify(Entity entity)
```

### main

```
public static void main(java.lang.String[] args)
```

(continued from last page)

## tud.iir.classification.entity Class Urns

java.lang.Object

└-tud.iir.classification.entity.Urns

---

public class **Urns**  
extends java.lang.Object

The URNS assessment model for (un)supervised information extraction. Single URN model with simplified assumptions (Poisson) as described in "A Probabilistic Model of Redundancy in Information Extraction, 2006". "Redundancy in Web-scale Information Extraction: Probabilistic Model and Experimental Results, 2008" (page 34 and following)

**Author:**

David Urbansky

---

### Constructors

#### Urns

public **Urns**()

### Methods

#### classify

public boolean **classify**([Entity](#) entity)

---

#### main

public static void **main**(java.lang.String[] args)

## tud.iir.classification.entity Class WordFeatureClassifier

```
java.lang.Object
├── tud.iir.classification.entity.EntityAssessor
│   └── tud.iir.classification.entity.WordFeatureClassifier
```

---

```
public class WordFeatureClassifier
extends EntityAssessor
```

The WordFeatureClassifier uses only string features from the entity name to assess or classify it.

**Author:**

David Urbansky

---

### Constructors

#### WordFeatureClassifier

```
public WordFeatureClassifier()
```

### Methods

#### main

```
public static void main(java.lang.String[] args)
```

**Parameters:**

args

---

Package

**tud.iir.classification.mio**

# tud.iir.classification.mio

## Class MIOClassifier

```
java.lang.Object
├── tud.iir.classification.Classifier
│   └── tud.iir.classification.mio.MIOClassifier
```

```
public class MIOClassifier
extends Classifier
```

The MIOClassifier calculate scores for ranking of MIOs. Attention: First train the Classifier, then save it. For classifying the classifier must be loaded first.

## Constructors

### MIOClassifier

```
public MIOClassifier()
```

Instantiates a new mIO classifier.

## Methods

### classify

```
public void classify(MIO mio)
```

Calculate the regression value for a given MIO.

**Parameters:**

`mio` - the MIO

**Returns:**

the float

### trainClassifier

```
public void trainClassifier(java.lang.String filePath)
```

Train classifier.

**Parameters:**

`filePath` - the file path

**Returns:**

true, if successful

### loadTrainedClassifier

```
public void loadTrainedClassifier()
```

Load an already trained classifier.

---

## saveTrainedClassifier

```
public void saveTrainedClassifier()
```

Simply save the trained classifier.

---

## doesTrainedMIOClassifierExists

```
public boolean doesTrainedMIOClassifierExists()
```

Check if an already trained MIOClassifier exists.

---

## main

```
public static void main(java.lang.String[] args)
```

The main method.

**Parameters:**

`args` - the arguments

---

Package

**tud.iir.classification.page**



# tud.iir.classification.page Class ClassificationDocument

java.lang.Object

└--tud.iir.classification.page.ClassificationDocument

Direct Known Subclasses:

[TestDocument](#)

---

```
public class ClassificationDocument
extends java.lang.Object
```

The document representation.

**Author:**

David Urbansky

---

## Fields

### TEST

```
public static final int TEST
```

Constant value: 1

---

### TRAINING

```
public static final int TRAINING
```

Constant value: 2

---

### UNCLASSIFIED

```
public static final int UNCLASSIFIED
```

Constant value: 3

## Constructors

### ClassificationDocument

```
public ClassificationDocument()
```

The constructor.

## Methods

### setRealCategories

```
public void setRealCategories(Categories categories)
```

Set the real categories (mainly for training documents).

(continued from last page)

**Parameters:**`categories` - The real categories.

---

## getRealCategories

```
public Categories getRealCategories()
```

Get the real categories of the document.

**Returns:**

The real categories.

---

## getRealCategoriesString

```
public java.lang.String getRealCategoriesString()
```

---

## getFirstRealCategory

```
public Category getFirstRealCategory()
```

---

## getUrl

```
public java.lang.String getUrl()
```

---

## setUrl

```
public void setUrl(java.lang.String url)
```

---

## getMainCategoryEntry

```
public CategoryEntry getMainCategoryEntry(boolean relevanceInPercent)
```

Get the category that is most relevant to this document.

**Parameters:**

`relevanceInPercent` - If true then the relevance will be output in percent.

**Returns:**

The most relevant category.

---

## getMainCategoryEntry

```
public CategoryEntry getMainCategoryEntry()
```

---

## sortCategoriesByRelevance

```
public void sortCategoriesByRelevance()
```

(continued from last page)

---

## getAssignedCategoryEntriesByRelevance

```
public CategoryEntries getAssignedCategoryEntriesByRelevance(int classType)
```

---

## getAssignedCategoryEntries

```
public CategoryEntries getAssignedCategoryEntries(boolean relevancesInPercent)
```

Get all categories for the document.

**Parameters:**

relevancesInPercent - If true then the relevance will be output in percent.

**Returns:**

All categories.

---

## getAssignedCategoryEntries

```
public CategoryEntries getAssignedCategoryEntries()
```

---

## getAssignedCategoryEntryNames

```
public java.lang.String getAssignedCategoryEntryNames()
```

---

## assignCategoryEntries

```
public void assignCategoryEntries(CategoryEntries categoryEntries)
```

---

## addCategoryEntry

```
public void addCategoryEntry(CategoryEntry categoryEntry)
```

---

## limitCategories

```
public void limitCategories(int minCategories,  
    int maxCategories,  
    double relevanceThreshold)
```

Limit number of assigned categories.

**Parameters:**

number - Number of categories to keep.

relevanceThreshold - Categories must have at least this much relevance to be kept.

(continued from last page)

## getWeightedTerms

```
public java.util.HashMap getWeightedTerms()
```

---

## setWeightedTerms

```
public void setWeightedTerms(java.util.HashMap weightedTerms)
```

---

## getDocumentType

```
public int getDocumentType()
```

---

## setDocumentType

```
public void setDocumentType(int documentType)
```

---

## getClassifiedAs

```
public int getClassifiedAs()
```

---

## getClassifiedAsReadable

```
public java.lang.String getClassifiedAsReadable()
```

---

## setClassifiedAs

```
public void setClassifiedAs(int classifiedAs)
```

---

## toString

```
public java.lang.String toString()
```

---

# tud.iir.classification.page

## Class ClassificationDocuments

```

java.lang.Object
  |-- java.util.AbstractCollection
        |-- java.util.AbstractList
              |-- java.util.ArrayList
                    |-- tud.iir.classification.page.ClassificationDocuments

```

### All Implemented Interfaces:

java.util.Collection, java.util.List, java.io.Serializable, java.lang.Cloneable,  
java.util.RandomAccess, java.util.List

```

public class ClassificationDocuments
extends java.util.ArrayList

```

An ArrayList of documents.

### Author:

David Urbansky

## Constructors

### ClassificationDocuments

```
public ClassificationDocuments()
```

## Methods

### getClassifiedNumberOfCategory

```
public int getClassifiedNumberOfCategory(java.lang.String categoryName)
```

Get the number of documents that have been assigned to given category.

#### Parameters:

categoryName - The name of the category.

#### Returns:

number The number of documents classified in the given category.

### getClassifiedNumberOfCategory

```
public int getClassifiedNumberOfCategory(Category category)
```

Get the number of documents that have been assigned to given category.

#### Parameters:

categoryName - The category.

#### Returns:

(continued from last page)

number The number of documents classified in the given category.

---

## getRealNumberOfCategory

```
public int getRealNumberOfCategory(java.lang.String categoryName)
```

Get the number of documents that actually ARE in the given category.

**Parameters:**

categoryName

**Returns:**

number

---

## getRealNumberOfCategory

```
public int getRealNumberOfCategory(Category category)
```

Get the number of documents that actually ARE in the given category.

**Parameters:**

category

**Returns:**

number

---

# tud.iir.classification.page

## Class ClassifierManager

java.lang.Object

└-tud.iir.classification.page.ClassifierManager

---

```
public class ClassifierManager
extends java.lang.Object
```

This class loads the training and test data, classifies and stores the results.

**Author:**

David Urbansky

---

## Constructors

### ClassifierManager

```
public ClassifierManager()
```

## Methods

### learnAndTestClassifierOnline

```
public final void learnAndTestClassifierOnline()
```

Retrieve web pages for a set of categories implying their category.

---

### trainAndTestClassifier

```
public final void trainAndTestClassifier(TextClassifier classifier,
EvaluationSetting evaluationSetting)
```

---

### trainClassifier

```
public final void trainClassifier(Dataset dataset,
TextClassifier classifier)
```

---

### testClassifier

```
public final void testClassifier(Dataset dataset,
TextClassifier classifier)
```

---

### log

```
public static void log(java.lang.String message)
```

---

---

## getTrainingDataPercentage

```
public final int getTrainingDataPercentage()
```

---

## setTrainingDataPercentage

```
public final void setTrainingDataPercentage(int trainingDataPercentage)
```

---

## load

```
public static TextClassifier load(java.lang.String classifierName)
```

---

## learnBestClassifier

```
public final void learnBestClassifier(java.util.List classificationTypeSettings,  
    java.util.List classifiers,  
    java.util.List featureSettings,  
    EvaluationSetting evaluationSetting)
```

This method simplifies the search for the best combination of classifier and feature settings. It automatically learns and evaluates all given combinations. The result will be a ranked list (by F1 score) of the combinations that perform best on the given training/test data.

### Parameters:

```
classificationTypeSettings  
featureSettings  
classifiers  
evaluationSetting
```

---

## main

```
public static void main(java.lang.String[] args)
```

If arguments are given, they must be in the following order: trainingPercentage inputFilePath classifierType classificationType training For example: java -jar classifierManager.jar 80 data/benchmarkSelection/page/deliciouspages\_cleansed\_400.txt 1 3 true

### Parameters:

```
args
```



# tud.iir.classification.page Class CombinedClassifier

```
java.lang.Object
├── tud.iir.classification.page.TextClassifier
│   ├── tud.iir.classification.page.DictionaryClassifier
│   │   └── tud.iir.classification.page.CombinedClassifier
```

---

**Deprecated.** *probably won't work anymore after refactoring*

```
public class CombinedClassifier
extends DictionaryClassifier
```

Combine URL and FullPage classification.

**Author:**

David Urbansky

---

## Constructors

### CombinedClassifier

```
public CombinedClassifier()
```

**Deprecated.**

## Methods

### preprocessDocument

```
public ClassificationDocument preprocessDocument(java.lang.String url)
```

**Deprecated.**

This method turns a web document into a document that can be classified. The subclasses implement this method according to the information they need for a classification document.

---

### preprocessDocument

```
public ClassificationDocument preprocessDocument(java.lang.String url,
ClassificationDocument classificationDocument)
```

**Deprecated.**

---

### main

```
public static void main(java.lang.String[] args)
```

**Deprecated.**

---

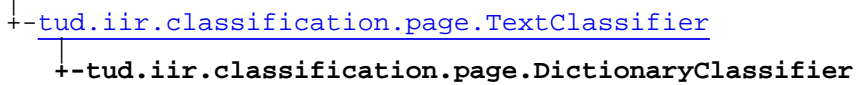
(continued from last page)

## Parameters:

args

# tud.iir.classification.page Class DictionaryClassifier

java.lang.Object



Direct Known Subclasses:

[CombinedClassifier](#), [FullPageClassifier](#), [URLClassifier](#)

---

```
public class DictionaryClassifier
extends TextClassifier
```

This classifier builds a weighed term look up table for the categories to classify new documents.

**Author:**

David Urbansky

---

## Constructors

### DictionaryClassifier

```
public DictionaryClassifier()
```

---

### DictionaryClassifier

```
public DictionaryClassifier(java.lang.String name)
```

---

## Methods

### init

```
public void init()
```

---

### useIndex

```
public void useIndex()
```

---

### useMemory

```
public void useMemory()
```

---

(continued from last page)

---

## addToDictionary

```
public void addToDictionary(ClassificationDocument trainingDocument,  
    int classType)
```

---

## save

```
public void save()
```

---

## saveDictionary

```
public void saveDictionary(boolean indexFirst,  
    boolean deleteIndexFirst)
```

Serialize the dictionary. All category information and parameters will be saved in the .ser file. The actual dictionary will be stored in the dictionary index.

### Parameters:

`classType` - The class type for the dictionary to distinguish the name.

---

## loadDictionary

```
public void loadDictionary()
```

---

## loadDictionary

```
public void loadDictionary(int classType)
```

Load the dictionary into memory, or activate it when several have been loaded.

---

## loadAllDictionaries

```
public void loadAllDictionaries()
```

Load all dictionaries into memory.

---

## classify

```
public ClassificationDocument classify(ClassificationDocument document,  
    boolean loadDictionary)
```

---

## classify

```
public ClassificationDocument classify(ClassificationDocument document)
```

This method is implemented in concrete classifiers.

---

## classifyTestDocuments

```
public void classifyTestDocuments(boolean loadDictionary)
```

(continued from last page)

---

## preprocessDocument

```
public ClassificationDocument preprocessDocument(java.lang.String text,  
        ClassificationDocument classificationDocument)
```

---

## preprocessDocument

```
public ClassificationDocument preprocessDocument(java.lang.String text)
```

This method turns a web document into a document that can be classified. The subclasses implement this method according to the information they need for a classification document.

---

## getDictionary

```
public Dictionary getDictionary()
```

---

## setDictionary

```
public void setDictionary(Dictionary dictionary)
```

# tud.iir.classification.page Class FullPageClassifier

```
java.lang.Object
├── tud.iir.classification.page.TextClassifier
│   ├── tud.iir.classification.page.DictionaryClassifier
│   └── tud.iir.classification.page.FullPageClassifier
```

Deprecated.

```
public class FullPageClassifier
extends DictionaryClassifier
```

## Constructors

### FullPageClassifier

```
public FullPageClassifier()
```

Deprecated.

## Methods

### preprocessDocument

```
public ClassificationDocument preprocessDocument(java.lang.String url)
```

Deprecated.

This method turns a web document into a document that can be classified. The subclasses implement this method according to the information they need for a classification document.

### preprocessDocument

```
public ClassificationDocument preprocessDocument(java.lang.String url,
ClassificationDocument classificationDocument)
```

Deprecated.

# tud.iir.classification.page

## Class KNNClassifier

java.lang.Object

└-[tud.iir.classification.page.TextClassifier](#)  
└-**tud.iir.classification.page.KNNClassifier**

All Implemented Interfaces:  
java.io.Serializable

---

public class **KNNClassifier**  
extends [TextClassifier](#)  
implements java.io.Serializable

a concrete KNN classifier

**Author:**  
David Urbansky

---

## Constructors

### KNNClassifier

public **KNNClassifier**()

The constructor.

## Methods

### classify

public [ClassificationDocument](#) **classify**([ClassificationDocument](#) document)

This method is implemented in concrete classifiers.

---

### save

public void **save**()

---

### getK

public int **getK**()

---

### setK

public void **setK**(int k)

---

(continued from last page)

## getParameters

```
public java.lang.String getParameters()
```

Get parameters used for the classifier (only k).

---

## preprocessDocument

```
public ClassificationDocument preprocessDocument(java.lang.String url)
```

This method turns a web document into a document that can be classified. The subclasses implement this method according to the information they need for a classification document.

---

## preprocessDocument

```
public ClassificationDocument preprocessDocument(java.lang.String url,  
        ClassificationDocument classificationDocument)
```



# tud.iir.classification.page

## Class NGram

java.lang.Object

└─ tud.iir.classification.page.NGram

All Implemented Interfaces:

java.io.Serializable

---

```
public class NGram
    extends java.lang.Object
    implements java.io.Serializable
```

An n-Gram.

Author:

David Urbansky

---

## Constructors

### NGram

```
public NGram(java.lang.String string)
```

## Methods

### getString

```
public java.lang.String getString()
```

---

### setString

```
public void setString(java.lang.String string)
```

---

### getN

```
public int getN()
```

---

### setN

```
public void setN(int n)
```

---

(continued from last page)

## getFrequency

```
public int getFrequency()
```

---

## setFrequency

```
public void setFrequency(int frequency)
```

---

## increaseFrequency

```
public void increaseFrequency()
```

---

## getIdf

```
public double getIdf(int documentCount)
```

---

## getIdf

```
public double getIdf()
```

---

## calculateIdf

```
public void calculateIdf(int documentCount)
```

---

## setIdf

```
public void setIdf(double idf)
```

---

## getIndex

```
public int getIndex()
```

---

## setIndex

```
public void setIndex(int index)
```

---

## toString

```
public java.lang.String toString()
```

---

(continued from last page)

## tud.iir.classification.page Class NGramIndex

```
java.lang.Object
├-- java.util.AbstractMap
│   └-- java.util.HashMap
│       └-- tud.iir.classification.page.NGramIndex
```

### All Implemented Interfaces:

java.io.Serializable, java.util.Map, java.io.Serializable, java.lang.Cloneable, java.util.Map, java.io.Serializable

---

```
public class NGramIndex
  extends java.util.HashMap
  implements java.util.Map, java.lang.Cloneable, java.io.Serializable, java.util.Map,
  java.io.Serializable
```

---

## Constructors

### NGramIndex

```
public NGramIndex()
```

## Methods

### put

```
public NGram put(java.lang.String key,
  NGram value)
```

### getNGram

```
public NGram getNGram(java.lang.String ngramString)
```

---

### getTop

```
public java.util.Set getTop(int k)
```

---

### getNumberOfDocuments

```
public int getNumberOfDocuments()
```

---

---

## setNumberOfDocuments

```
public void setNumberOfDocuments(int numberOfDocuments)
```

---

## increasNumberOfDocuments

```
public void increasNumberOfDocuments()
```

# tud.iir.classification.page

## Class Preprocessor

```
java.lang.Object
├--tud.iir.classification.page.Preprocessor
```

```
public final class Preprocessor
extends java.lang.Object
```

The preprocessor reads the terms for a given resource and weights them according to their relevance. 2010-06-09, Philipp, added [preProcessText\(String\)](#) and [preProcessText\(String, ClassificationDocument\)](#)

**Author:**  
David Urbansky, Philipp Katz

## Fields

### WEIGHT\_DOMAIN\_TERM

```
public static final double WEIGHT_DOMAIN_TERM
```

Constant value: 8.0

### WEIGHT\_TITLE\_TERM

```
public static final double WEIGHT_TITLE_TERM
```

Constant value: 7.0

### WEIGHT\_KEYWORD\_TERM

```
public static final double WEIGHT_KEYWORD_TERM
```

Constant value: 6.0

### WEIGHT\_META\_TERM

```
public static final double WEIGHT_META_TERM
```

Constant value: 4.0

### WEIGHT\_BODY\_TERM

```
public static final double WEIGHT_BODY_TERM
```

Constant value: 1.0

## Constructors

(continued from last page)

## Preprocessor

```
public Preprocessor(TextClassifier classifier)
```

## Methods

### preProcessDocument

```
public ClassificationDocument preProcessDocument(java.lang.String inputString,  
ClassificationDocument classificationDocument)
```

Preprocess a string (such as a URL) and create a classification document. A map of n-grams is created for the document and added to it. If a n-gram term exists, it will be taken from the n-gram index.

**Parameters:**

inputString - The input string.

classificationDocument - The classification document.

**Returns:**

The classification document with the n-gram map.

---

### preProcessDocument

```
public ClassificationDocument preProcessDocument(java.lang.String url)
```

---

### preProcessString

```
public ClassificationDocument preProcessString(java.lang.String inputString,  
ClassificationDocument classificationDocument)
```

**Deprecated.** *consider using preprocess document*

Preprocess a string (such as a URL) and create a classification document. A map of n-grams is created for the document and added to it. If a n-gram term exists, it will be taken from the n-gram index.

**Parameters:**

inputString - The input string.

classificationDocument - The classification document.

**Returns:**

The classification document with the n-gram map.

---

### preProcessString

```
public ClassificationDocument preProcessString(java.lang.String url)
```

---

### preProcessPage

```
public ClassificationDocument preProcessPage(java.lang.String url,  
ClassificationDocument classificationDocument)
```

(continued from last page)

**Deprecated.** *consider using preprocess document*

Preprocess a web page and create a classification document. A map of terms is created for the document and added to it. If a term exists, it will be taken from the term index.

**Parameters:**

url - The URL of the web page.

classificationDocument - The classification document.

**Returns:**

The classification document with the n-gram map.

---

## preProcessPage

```
public ClassificationDocument preProcessPage(java.lang.String url)
```

**Deprecated.** *consider using preprocess document*

**Parameters:**

url

**Returns:**

---

## preProcessText

```
public ClassificationDocument preProcessText(java.lang.String text,  
      ClassificationDocument classificationDocument)
```

**Deprecated.** *consider using preprocess document*

Preprocesses a long string of text similar to [preProcessPage\(String, ClassificationDocument\)](#), but the text content is not downloaded from the web but passed via the url parameter. XXX This is a quick and dirty hack to allow classification of text content and should be refactored somehow in the future.

**Parameters:**

text - the text to be preProcessed

classificationDocument

**Returns:**

---

## preProcessText

```
public ClassificationDocument preProcessText(java.lang.String text)
```

**Deprecated.** *consider using preprocess document*

**Parameters:**

text

**Returns:**

---



# tud.iir.classification.page

## Class TestDocument

```
java.lang.Object
├── tud.iir.classification.page.ClassificationDocument
│   └── tud.iir.classification.page.TestDocument
```

```
public class TestDocument
    extends ClassificationDocument
```

A test document is a document that has given information about the correct category but is classified using a classifier. It is used to determine the accuracy of the classifier.

**Author:**

David Urbansky

## Constructors

### TestDocument

```
public TestDocument()
```

## Methods

### getCorrectlyAssignedCategoryEntries

```
public CategoryEntries getCorrectlyAssignedCategoryEntries()
```

### getPrecisionAt

```
public double getPrecisionAt(int rank)
```

### isCorrectClassified

```
public boolean isCorrectClassified()
```

Returns true if the document is correct classified. Hierarchical classified documents count as correct if main category matches. Tag classified documents count as correct if first (main) tags matches any real tag.

**Returns:**

True if the document is correct classified, false otherwise.

# tud.iir.classification.page Class TextClassifier

java.lang.Object

└--tud.iir.classification.page.TextClassifier

Direct Known Subclasses:

[DictionaryClassifier](#), [KNNClassifier](#)

---

public abstract class **TextClassifier**  
extends java.lang.Object

The classifier is an abstract class that provides basic methods used by concrete classifiers.

**Author:**

David Urbansky

---

## Fields

### categories

public tud.iir.classification.Categories **categories**

A classifier classifies to certain categories.

## Constructors

### TextClassifier

public **TextClassifier**()

The constructor, initiate members.

## Methods

### reset

public void **reset**()

Reset the classifier.

---

### getCategories

public [Categories](#) **getCategories**()

**Returns:**

All the categories the classifier orders documents to.

---

### setCategories

public void **setCategories**([Categories](#) categories)

(continued from last page)

**Parameters:**

`categories` - All the categories the classifier orders documents to.

---

**getTrainingDocuments**

```
public ClassificationDocuments getTrainingDocuments()
```

---

**setTrainingDocuments**

```
public void setTrainingDocuments(ClassificationDocuments trainingDocuments)
```

---

**getTestDocuments**

```
public ClassificationDocuments getTestDocuments()
```

---

**setTestDocuments**

```
public void setTestDocuments(ClassificationDocuments testDocuments)
```

---

**getPreprocessor**

```
public Preprocessor getPreprocessor()
```

---

**setPreprocessor**

```
public void setPreprocessor(Preprocessor preprocessor)
```

---

**isBenchmark**

```
public boolean isBenchmark()
```

---

**setBenchmark**

```
public void setBenchmark(boolean benchmark)
```

---

**isForum**

```
public static boolean isForum(java.lang.String url)
```

Check whether a given web page is a forum/board page. Make use of heuristics.

---

(continued from last page)

**Parameters:**

`url` - The url of the web page.

**Returns:**

True if it is considered a forum, false otherwise.

---

## isForum

```
public static boolean isForum(org.w3c.dom.Document document)
```

**Parameters:**

`document`

**Returns:**

---

## isFAQ

```
public static boolean isFAQ(java.lang.String url)
```

Check whether given url is a FAQ web page. Make use of heuristics.

**Parameters:**

`url` - The url of the page.

**Returns:**

True if the page has FAQ, false otherwise.

---

## isFAQ

```
public static boolean isFAQ(org.w3c.dom.Document document)
```

---

## classifyTestDocuments

```
public void classifyTestDocuments()
```

this method calls the classify function that is implemented by each concrete classifier all test documents are classified

---

## preprocessDocument

```
public abstract ClassificationDocument preprocessDocument(java.lang.String text)
```

This method turns a web document into a document that can be classified. The subclasses implement this method according to the information they need for a classification document.

**Parameters:**

`document` - The web document that should be prepared for classification.

**Returns:**

A document that can be classified.

---

---

## preprocessDocument

```
public abstract ClassificationDocument preprocessDocument(java.lang.String text,  
    ClassificationDocument classificationDocument)
```

---

## classify

```
public ClassificationDocument classify(java.lang.String text)
```

Classify a document that is given with an URL. This method is implemented in concrete classifiers.

**Parameters:**

`url` - The URL of the document that has to be classified.

**Returns:**

A classified document.

---

---

## classify

```
public abstract ClassificationDocument classify(ClassificationDocument document)
```

This method is implemented in concrete classifiers.

**Parameters:**

`document` - The document that has to be classified.

**Returns:**

A classified document.

---

---

## getParameters

```
public java.lang.String getParameters()
```

Get the parameters used for the classifier.

**Returns:**

A string with information about the parameters that have been set for the classifier.

---

---

## getName

```
public java.lang.String getName()
```

---

## setName

```
public void setName(java.lang.String name)
```

---

## setClassificationTypeSetting

```
public void setClassificationTypeSetting(ClassificationTypeSetting  
classificationTypeSetting)
```

---

## getClassificationTypeSetting

```
public ClassificationTypeSetting getClassificationTypeSetting()
```

---

## getClassificationType

```
public int getClassificationType()
```

---

## isSerialize

```
public boolean isSerialize()
```

---

## setFeatureSetting

```
public void setFeatureSetting(FeatureSetting featureSetting)
```

---

## getFeatureSetting

```
public FeatureSetting getFeatureSetting()
```

---

## setPerformance

```
public void setPerformance(ClassifierPerformance performance)
```

---

## getPerformance

```
public ClassifierPerformance getPerformance()
```

---

## getPerformanceCopy

```
public ClassifierPerformance getPerformanceCopy()
```

Get a copy of the classifier performance. Delete weighted terms in documents to lower memory consumption.

**Returns:**

A new instance of classifier performance.

---

## showTrainingDocuments

```
public java.lang.String showTrainingDocuments()
```

---

---

## showTestDocuments

```
public java.lang.String showTestDocuments()
```

XXX TextClassifier line 380, calculation must be the same, CrossValidator && console output, see mail Philipp to David [mail](#)

---

## toString

```
public java.lang.String toString()
```

---

## save

```
public abstract void save()
```

# tud.iir.classification.page

## Class URLClassifier

```

java.lang.Object
├── tud.iir.classification.page.TextClassifier
│   ├── tud.iir.classification.page.DictionaryClassifier
│   └── tud.iir.classification.page.URLClassifier

```

**Deprecated.** *Use this class maybe with preset feature and classification type settings*

```

public class URLClassifier
extends DictionaryClassifier

```

Classify a web page only by its URL. Implementation similar to the one described in "Purely URL-based Topic Classification, 2009".

**Author:**

David Urbansky TODO inherit from classifier and use createInstance etc. from there

## Constructors

### URLClassifier

```
public URLClassifier()
```

**Deprecated.**

## Methods

### preprocessDocument

```
public ClassificationDocument preprocessDocument(java.lang.String url)
```

**Deprecated.**

This method turns a web document into a document that can be classified. The subclasses implement this method according to the information they need for a classification document.

### preprocessDocument

```
public ClassificationDocument preprocessDocument(java.lang.String url,
ClassificationDocument classificationDocument)
```

**Deprecated.**

### main

```
public static void main(java.lang.String[] args)
```

**Deprecated.**



(continued from last page)

**Parameters:**

args

## tud.iir.classification.page Class URLs

```
java.lang.Object
  |-- java.util.AbstractCollection
        |-- java.util.AbstractList
              |-- java.util.ArrayList
                    |-- tud.iir.classification.page.URLs
```

### All Implemented Interfaces:

```
java.util.Collection, java.util.List, java.io.Serializable, java.lang.Cloneable,
java.util.RandomAccess, java.util.List
```

---

```
public class URLs
extends java.util.ArrayList
```

an ArrayList of URLs

**Author:**  
David Urbansky

---

## Constructors

### URLs

```
public URLs()
```

---

Package

**tud.iir.classification.page.evaluation**

# tud.iir.classification.page.evaluation Class AverageClassifierPerformance

java.lang.Object

└--tud.iir.classification.page.evaluation.AverageClassifierPerformance

---

public class **AverageClassifierPerformance**  
extends java.lang.Object

This class is a container for the averaged classifier performance.

**Author:**

David Urbansky

---

## Constructors

### AverageClassifierPerformance

public **AverageClassifierPerformance**()

## Methods

### getPrecision

public double **getPrecision**()

### setPrecision

public void **setPrecision**(double precision)

### getRecall

public double **getRecall**()

### setRecall

public void **setRecall**(double recall)

### getF1

public double **getF1**()

Calculate the F1 score.

---

(continued from last page)

**Returns:**

The F1 score.

# tud.iir.classification.page.evaluation Class ClassificationTypeSetting

java.lang.Object

└--tud.iir.classification.page.evaluation.ClassificationTypeSetting

```
public class ClassificationTypeSetting
extends java.lang.Object
```

The settings which classification type and which settings for that should be used for a classifier.

**Author:**

David Urbansky

## Fields

### SINGLE

```
public static final int SINGLE
```

Take only the first category specified in the txt file.  
Constant value: 1

### HIERARCHICAL

```
public static final int HIERARCHICAL
```

Take all categories and treat them as a hierarchy.  
Constant value: 2

### TAG

```
public static final int TAG
```

Take all categories and treat them as tags.  
Constant value: 3

## Constructors

### ClassificationTypeSetting

```
public ClassificationTypeSetting()
```

## Methods

### setClassificationType

```
public void setClassificationType(int classificationType)
```

Set the classification type under which the classifier operates.

**Parameters:**

(continued from last page)

`classificationType` - The classification type must be one of `TextClassifier.SINGLE`, `TextClassifier.HIERARCHICAL`, or `TextClassifier.TAG`.

---

## getClassificationType

```
public int getClassificationType()
```

---

## setClassificationTypeTagSetting

```
public void setClassificationTypeTagSetting(ClassificationTypeTagSetting
classificationTypeTagSetting)
```

---

## getClassificationTypeTagSetting

```
public ClassificationTypeTagSetting getClassificationTypeTagSetting()
```

---

## setSerializeClassifier

```
public void setSerializeClassifier(boolean serializeClassifier)
```

---

## isSerializeClassifier

```
public boolean isSerializeClassifier()
```

---

## toString

```
public java.lang.String toString()
```

# tud.iir.classification.page.evaluation

## Class ClassificationTypeTagSetting

java.lang.Object

└--tud.iir.classification.page.evaluation.ClassificationTypeTagSetting

---

public class **ClassificationTypeTagSetting**  
extends java.lang.Object

More specific settings for the `ClassificationTypeSetting.TAG` setting.

**Author:**

David Urbansky

---

## Constructors

### ClassificationTypeTagSetting

public **ClassificationTypeTagSetting**()

## Methods

### getTagConfidenceThreshold

public double **getTagConfidenceThreshold**()

### setTagConfidenceThreshold

public void **setTagConfidenceThreshold**(double tagConfidenceThreshold)

### getMinTags

public int **getMinTags**()

### setMinTags

public void **setMinTags**(int minTags)

### getMaxTags

public int **getMaxTags**()



---

## setMaxTags

```
public void setMaxTags(int maxTags)
```

---

## isTagBoost

```
public boolean isTagBoost()
```

---

## setTagBoost

```
public void setTagBoost(boolean tagBoost)
```

---

## isUseCooccurrence

```
public boolean isUseCooccurrence()
```

---

## setUseCooccurrence

```
public void setUseCooccurrence(boolean useCooccurrence)
```

---

## toString

```
public java.lang.String toString()
```

# tud.iir.classification.page.evaluation

## Class ClassifierPerformance

java.lang.Object

└--tud.iir.classification.page.evaluation.ClassifierPerformance

public class **ClassifierPerformance**  
extends java.lang.Object

This class calculates scores for a given classifier such as precision, recall, and F1.

**Author:**

David Urbansky

## Constructors

### ClassifierPerformance

public **ClassifierPerformance**([TextClassifier](#) classifier)

Create a new ClassifierPerformance for a given classifier.

**Parameters:**

classifier - The classifier.

## Methods

### getCategories

public [Categories](#) **getCategories**()

### setCategories

public void **setCategories**([Categories](#) categories)

### setClassificationType

public void **setClassificationType**(int classificationType)

### getClassificationType

public int **getClassificationType**()

### setTrainingDocuments

public void **setTrainingDocuments**([ClassificationDocuments](#) trainingDocuments)

(continued from last page)

---

## getTrainingDocuments

```
public ClassificationDocuments getTrainingDocuments()
```

---

## setTestDocuments

```
public void setTestDocuments(ClassificationDocuments testDocuments)
```

---

## getTestDocuments

```
public ClassificationDocuments getTestDocuments()
```

---

## getNumberOfCorrectClassifiedDocumentsInCategory

```
public int getNumberOfCorrectClassifiedDocumentsInCategory(Category category)
```

Get the number of correct classified documents in a given category.

**Parameters:**

category - The category.

**Returns:**

Number of correct classified documents in a given category.

---

## getPrecisionForCategory

```
public double getPrecisionForCategory(Category category)
```

calculate and return the precision for a given category

**Parameters:**

category

**Returns:**

the precision for a given category

---

## getRecallForCategory

```
public double getRecallForCategory(Category category)
```

calculate and return the recall for a given category

**Parameters:**

category

**Returns:**

the recall for a given category

---

(continued from last page)

## getFForCategory

```
public double getFForCategory(Category category,  
    double alpha)
```

Calculate and return the F for a given category.

### Parameters:

`category` - The category.

`alpha` - A value between 0 and 1 to weight precision and recall (0.5 for F1).

### Returns:

F for a given category.

---

## getSensitivityForCategory

```
public double getSensitivityForCategory(Category category)
```

Calculate the sensitivity for a given category. Sensitivity =  $TP / (TP + FN)$ . Sensitivity specifies what percentage of actual category members were found. 100% sensitivity means that all actual documents belonging to the category were classified correctly.

### Parameters:

`category`

### Returns:

---

## getSpecificityForCategory

```
public double getSpecificityForCategory(Category category)
```

Calculate the specificity for a given category. Specificity =  $TN / (TN + FP)$ . Specificity specifies what percentage of not-category members were recognized as such. 100% specificity means that there were no documents classified as category member when they were actually not.

### Parameters:

`category` - The category.

### Returns:

The specificity.

---

## getAccuracyForCategory

```
public double getAccuracyForCategory(Category category)
```

Calculate the accuracy for a given category. Accuracy =  $(TP + TN) / (TP + TN + FP + FN)$ .

### Parameters:

`category` - The category.

### Returns:

The accuracy.

---

## getWeightForCategory

```
public double getWeightForCategory(Category category)
```

(continued from last page)

Calculate the prior for the given category. The prior is determined by calculating the frequency of the category in the training and test set and dividing it by the total number of documents. XXX use only test documents to determine prior?

**Parameters:**

`category` - The category for which the prior should be determined.

**Returns:**

The prior for the category.

---

## getAveragePrecision

```
public double getAveragePrecision(boolean weighted)
```

Get the average precision of all categories.

**Returns:**

The average precision of all categories.

---

## getAverageRecall

```
public double getAverageRecall(boolean weighted)
```

Get the average recall of all categories.

**Returns:**

The average recall of all categories.

---

## getAverageF

```
public double getAverageF(double alpha,  
    boolean weighted)
```

Get the average F of all categories.

**Parameters:**

`alpha` - to weight precision and recall (0.5 for F1)

**Returns:**

The average F of all categories.

---

## getAverageSensitivity

```
public double getAverageSensitivity(boolean weighted)
```

Calculate the average sensitivity.

**Parameters:**

`weighted` - If true, the average sensitivity is weighted using the priors of the categories.

**Returns:**

The (weighted) average sensitivity.

---

## getAverageSpecificity

```
public double getAverageSpecificity(boolean weighted)
```

Calculate the average specificity.

---

(continued from last page)

**Parameters:**

`weighted` - If true, the average accuracy is weighted using the priors of the categories.

**Returns:**

The (weighted) average accuracy.

---

## getAverageAccuracy

```
public double getAverageAccuracy(boolean weighted)
```

Calculate the average accuracy.

**Parameters:**

`weighted` - If true, the average accuracy is weighted using the priors of the categories.

**Returns:**

The (weighted) average accuracy.

# tud.iir.classification.page.evaluation

## Class CrossValidationResult

java.lang.Object

└-tud.iir.classification.page.evaluation.CrossValidationResult

public class **CrossValidationResult**  
extends java.lang.Object

The result of a cross validation for a classifier and given settings.

**Author:**

David Urbansky

## Constructors

### CrossValidationResult

public **CrossValidationResult**([TextClassifier](#) classifier)

## Methods

### setClassifier

public void **setClassifier**([TextClassifier](#) classifier)

### getClassifier

public [TextClassifier](#) **getClassifier**()

### setClassificationTypeSetting

public void **setClassificationTypeSetting**([ClassificationTypeSetting](#) classificationTypeSettings)

### getClassificationTypeSetting

public [ClassificationTypeSetting](#) **getClassificationTypeSetting**()

### setFeatureSetting

public void **setFeatureSetting**([FeatureSetting](#) featureSettings)

---

## getFeatureSetting

```
public FeatureSetting getFeatureSetting()
```

---

---

## getPerformancesDatasetTrainingFolds

```
public java.util.Set getPerformancesDatasetTrainingFolds()
```

---

---

## setPerformancesDatasetTrainingFolds

```
public void setPerformancesDatasetTrainingFolds(java.util.Set  
performancesDatasetTrainingFolds)
```

---

---

## getPerformancesTrainingFolds

```
public java.util.Map getPerformancesTrainingFolds()
```

---

---

## setPerformancesTrainingFolds

```
public void setPerformancesTrainingFolds(java.util.Map performancesTrainingFolds)
```

---

---

## getPerformancesFolds

```
public java.util.Map getPerformancesFolds()
```

---

---

## setPerformancesFolds

```
public void setPerformancesFolds(java.util.Map performancesFolds)
```

---

---

## getAveragePerformanceDataSetTrainingFolds

```
public AverageClassifierPerformance getAveragePerformanceDataSetTrainingFolds()
```

Calculate the average classifier performance when all performances over all datasets, training percentages, and folds are averaged.

**Returns:**

An average classifier performance.

---

---

## getAveragePerformanceTrainingFolds

```
public java.util.Map getAveragePerformanceTrainingFolds()
```

---



(continued from last page)

Calculate the average classifier performance when all performances over all training percentages and folds are averaged.

**Returns:**

The average classifier performance for each dataset.

---

## getAveragePerformanceFolds

```
public java.util.Map getAveragePerformanceFolds()
```

Calculate the average classifier performance when all performances over all folds are averaged.

**Returns:**

The average classifier performance for each dataset and training percentage.

# tud.iir.classification.page.evaluation

## Class CrossValidator

```
java.lang.Object
```

```
└─tud.iir.classification.page.evaluation.CrossValidator
```

```
public class CrossValidator
extends java.lang.Object
```

The CrossValidator validates a given classifier with the evaluation settings. It can also print results for manual investigation.

**Author:**

David Urbansky, Sandro Reichert

## Constructors

### CrossValidator

```
public CrossValidator()
```

## Methods

### setEvaluationSetting

```
public void setEvaluationSetting(EvaluationSetting evaluationSetting)
```

### getEvaluationSetting

```
public EvaluationSetting getEvaluationSetting()
```

### crossValidate

```
public CrossValidationResult crossValidate(TextClassifier classifier)
```

Method to compare the open analytix performance for classification depending on values trainingPercentage, threshold for assigning a second category and number of loops to average the performance with fixed trainingPercentage and threshold but random select of lines to be assigned to training and testing set

#### Parameters:

trainingPercentageMin - The percentage of the data set to be used for training - minimum value of loop, range [0,100].  
 trainingPercentageMax - The percentage of the data set to be used for training - maximum value of loop, range [0,100].  
 trainingPercentageStep - The percentage of the data set to be used for training - step between loops, range [0,100].  
 randomSplitTrainingDataSet - If true, initial data set is split randomly into training and test set (fixed percentage but randomly chosen lines). If false, the first lines are training set and the remainder is the test set.

(continued from last page)

`numberLoopsToAverage` - Number of loops to average the performance with fixed `trainingPercentage` and threshold but random select of lines to be assigned to training and testing set. Ignored if `randomSplitTrainingDataSet=false`, e.g. only one loop is executed per `trainingPercentage` and threshold.  
`thMin` - Minimum value for the threshold used to assign a second category.  
`thMax` - Maximum value for the threshold used to assign a second category.  
`thStep` - Value to add to the threshold per loop.  
`classType` - The type of `WebPageClassifier` to be used, e.g. `WebPageClassifier.FIRST`.

---

## printEvaluationFiles

```
public void printEvaluationFiles(java.util.Set cvResults,  
    java.lang.String outputFolder)
```

Print the evaluation files where a user can find out which classifier under which settings is the best for the given datasets. Three files will be written, one where each classifier's performance will be averaged over all datasets, training percentages, and folds. Another one where each classifier is only averaged over all training percentages and folds and a last one where each classifier is only averaged over all folds for a given dataset and training percentage.

### Parameters:

`cvResults` - A set of cross validation results.  
`outputFolder` - The path to the folder where the evaluation files should be written to.

# tud.iir.classification.page.evaluation Class Dataset

```
java.lang.Object
└--tud.iir.classification.page.evaluation.Dataset
```

---

```
public class Dataset
extends java.lang.Object
```

A simple representation of a dataset.

**Author:**  
David Urbansky

---

## Constructors

### Dataset

```
public Dataset()
```

## Methods

### setPath

```
public void setPath(java.lang.String path)
```

---

### getPath

```
public java.lang.String getPath()
```

---

### setSeparationString

```
public void setSeparationString(java.lang.String separationString)
```

---

### getSeparationString

```
public java.lang.String getSeparationString()
```

---

### toString

```
public java.lang.String toString()
```

---

# tud.iir.classification.page.evaluation

## Class EvaluationSetting

java.lang.Object

└--tud.iir.classification.page.evaluation.EvaluationSetting

---

public final class **EvaluationSetting**  
extends java.lang.Object

Set the evaluation settings for a classifier.

**Author:**

David Urbansky

---

## Fields

### PRESET\_SIMPLE\_EVALUATION

public static final int **PRESET\_SIMPLE\_EVALUATION**

Evaluate quickly.  
Constant value: 1

---

### PRESET\_MODERATE\_EVALUATION

public static final int **PRESET\_MODERATE\_EVALUATION**

Evaluate moderately.  
Constant value: 2

---

### PRESET\_INTENSE\_EVALUATION

public static final int **PRESET\_INTENSE\_EVALUATION**

Evaluate intensively.  
Constant value: 3

---

## Constructors

### EvaluationSetting

public **EvaluationSetting**()

In case no preset is chosen the empty constructor is called. All settings have to be made manually.

---

### EvaluationSetting

public **EvaluationSetting**(int preset)

---

## Methods

---

(continued from last page)

## getkFolds

```
public int getkFolds()
```

---

## setkFolds

```
public void setkFolds(int kFolds)
```

---

## isRandom

```
public boolean isRandom()
```

---

## setRandom

```
public void setRandom(boolean random)
```

---

## setTrainingPercentageMin

```
public void setTrainingPercentageMin(double trainingPercentageMin)
```

---

## getTrainingPercentageMin

```
public double getTrainingPercentageMin()
```

---

## setTrainingPercentageMax

```
public void setTrainingPercentageMax(double trainingPercentageMax)
```

---

## getTrainingPercentageMax

```
public double getTrainingPercentageMax()
```

---

## setTrainingPercentageStep

```
public void setTrainingPercentageStep(double trainingPercentageStep)
```

---

## getTrainingPercentageStep

```
public double getTrainingPercentageStep()
```

---

(continued from last page)

---

## setDatasets

```
public void setDatasets(java.util.List datasets)
```

---

## getDatasets

```
public java.util.List getDatasets()
```

---

## addDataset

```
public void addDataset(Dataset dataset)
```

---

## toString

```
public java.lang.String toString()
```

# tud.iir.classification.page.evaluation

## Class FeatureSetting

java.lang.Object

└--tud.iir.classification.page.evaluation.FeatureSetting

---

public class **FeatureSetting**  
extends java.lang.Object

Save the settings which text features should be used for a classifier.

**Author:**

David Urbansky

---

## Fields

### CHAR\_NGRAMS

public static final int **CHAR\_NGRAMS**

Use n-Grams on a character level.  
Constant value: 1

---

### WORD\_NGRAMS

public static final int **WORD\_NGRAMS**

Use n-Grams on a word level.  
Constant value: 2

---

### englishStopWords

public static java.util.Set **englishStopWords**

Set of English stop words.

---

## Constructors

### FeatureSetting

public **FeatureSetting**()

## Methods

### getTextFeatureType

public int **getTextFeatureType**()

---



(continued from last page)

## setTextFeatureType

```
public void setTextFeatureType(int textFeatureType)
```

---

## setMaxTerms

```
public void setMaxTerms(int maxTerms)
```

---

## getMaxTerms

```
public int getMaxTerms()
```

---

## getMinNGramLength

```
public int getMinNGramLength()
```

---

## setMinNGramLength

```
public void setMinNGramLength(int minNGramLength)
```

---

## getMaxNGramLength

```
public int getMaxNGramLength()
```

---

## setMaxNGramLength

```
public void setMaxNGramLength(int maxNGramLength)
```

---

## setMaximumTermLength

```
public void setMaximumTermLength(int maximumTermLength)
```

Set the maximum length of a single term, this only applies if `textFeatureType` is set to `WORD_NGRAMS` and `maxNGramLength` is 1, that is, only unigrams will be used.

---

## getMaximumTermLength

```
public int getMaximumTermLength()
```

---

## setMinimumTermLength

```
public void setMinimumTermLength(int minimumTermLength)
```

---

(continued from last page)

Set the minimum length of a single term, this only applies if `textFeatureType` is set to `WORD_NGRAMS` and `maxNGramLength` is 1, that is, only unigrams will be used.

---

## getMinimumTermLength

```
public int getMinimumTermLength()
```

---

## getStopWords

```
public java.util.Set getStopWords()
```

---

## setStopWords

```
public void setStopWords(java.util.Set stopWords)
```

---

## toString

```
public java.lang.String toString()
```

---

# tud.iir.classification.page.evaluation

## Class TrainingDataSeparation

java.lang.Object

└--tud.iir.classification.page.evaluation.TrainingDataSeparation

```
public class TrainingDataSeparation
extends java.lang.Object
```

This class separates a given training set into a training set and a evaluation set.

**Author:**

Sandro Reichert

## Constructors

### TrainingDataSeparation

```
public TrainingDataSeparation()
```

## Methods

### separateFile

```
public void separateFile(java.lang.String fileToSeparate,
    java.lang.String trainingDataFileToWrite,
    java.lang.String testingDataFileToWrite,
    double trainingDataPercentage,
    boolean randomlyChooseLines)
throws java.io.FileNotFoundException,
    java.io.IOException
```

Separates a given training set by trainingDataPercentage into two files, containing training and testing data. The separation can be done by randomly chosen lines or the first part is used for training and the second part for testing.

Example:

1) fileToSeparate contains 10 lines, trainingDataPercentage = 40 and randomlyChooseLines is false, than lines 1-4 are written to trainingDataFileToWrite and lines 5-10 are written to testingDataFileToWrite.

2) fileToSeparate contains 10 lines, trainingDataPercentage = 40 and randomlyChooseLines is true, than 4 randomly chosen lines are written to trainingDataFileToWrite and the remaining lines are written to testingDataFileToWrite.

#### Parameters:

fileToSeparate - Path to the file to be separated.

trainingDataFileToWrite - Path to the file the training data will be written to.

testingDataFileToWrite - Path to the file the testing data will be written to.

trainingDataPercentage - Percentage of file which should be used for training, range [0, 100]. The remainder of the file can be used for testing.

randomlyChooseLines - Specifies whether lines should be picked randomly or not. If false, the first lines are used for training.

#### Throws:

IllegalArgumentException - if trainingDataPercentage is out of range [0, 100].

FileNotFoundException - if fileToSeparate can not be found.

(continued from last page)

`IOException` - if `fileToSeparate` can not be accessed.

---

Package

**tud.iir.classification.qa**

# tud.iir.classification.qa

## Class AnswerClassifier

```
java.lang.Object
├── tud.iir.classification.Classifier
│   └── tud.iir.classification.qa.AnswerClassifier
```

---

```
public class AnswerClassifier
extends Classifier
```

Classify an answer for a question.

**Author:**

David Urbansky

---

## Constructors

### AnswerClassifier

```
public AnswerClassifier(int type)
```

## Methods

### useTrainedClassifier

```
public void useTrainedClassifier()
```

Use an already trained classifier. TODO pull this method up? I have copied this to NewsRankingClassifier for now. We should have the possibility to set file names for the serialized model to avoid conflicts between different Classifier subclasses -- Philipp.

---

### trainClassifier

```
public void trainClassifier(java.lang.String dirPath)
```

Train and save a classifier. Use all html documents in the specified path.

---

### testClassifier

```
public void testClassifier(java.lang.String dirPath)
```

---

### rankAnswer

```
public double rankAnswer(AnswerFeatures af)
```

---

### main

```
public static void main(java.lang.String[] args)
```

---

(continued from last page)

**Parameters:**

args

## tud.iir.classification.qa

### Class AnswerFeatures

```
java.lang.Object
└--tud.iir.classification.qa.AnswerFeatures
```

---

```
public class AnswerFeatures
extends java.lang.Object
```

---

## Constructors

### AnswerFeatures

```
public AnswerFeatures()
```

## Methods

### getAsFeatureObject

```
public FeatureObject getAsFeatureObject(int correct)
```

---

### getAnswerWordCount

```
public int getAnswerWordCount()
```

---

### setAnswerWordCount

```
public void setAnswerWordCount(int answerWordCount)
```

---

### getSimilarity1

```
public float getSimilarity1()
```

---

### setSimilarity1

```
public void setSimilarity1(float similarity1)
```

---



(continued from last page)

## getSimilarity2

```
public float getSimilarity2()
```

---

## setSimilarity2

```
public void setSimilarity2(float similarity2)
```

---

## getSimilarity3

```
public float getSimilarity3()
```

---

## setSimilarity3

```
public void setSimilarity3(float similarity3)
```

---

## getSimilarity4

```
public float getSimilarity4()
```

---

## setSimilarity4

```
public void setSimilarity4(float similarity4)
```

---

## getSimilarity5

```
public float getSimilarity5()
```

---

## setSimilarity5

```
public void setSimilarity5(float similarity5)
```

---

## getSimilarity6

```
public float getSimilarity6()
```

---

## setSimilarity6

```
public void setSimilarity6(float similarity6)
```

---

(continued from last page)

---

## getSimilarity7

```
public float getSimilarity7()
```

---

## setSimilarity7

```
public void setSimilarity7(float similarity7)
```

---

## getSimilarity8

```
public float getSimilarity8()
```

---

## setSimilarity8

```
public void setSimilarity8(float similarity8)
```

---

## isAnswerHintBeforeAnswer

```
public int isAnswerHintBeforeAnswer()
```

---

## setAnswerHintBeforeAnswer

```
public void setAnswerHintBeforeAnswer(int answerHintBeforeAnswer)
```

---

## getTagDistance

```
public int getTagDistance()
```

---

## setTagDistance

```
public void setTagDistance(int tagDistance)
```

---

## getWordDistance

```
public int getWordDistance()
```

---

(continued from last page)

## setWordDistance

```
public void setWordDistance(int wordDistance)
```

---

## getTagCount

```
public int getTagCount()
```

---

## setTagCount

```
public void setTagCount(int tagCount)
```

---

## getDistinctTagCount

```
public int getDistinctTagCount()
```

---

## setDistinctTagCount

```
public void setDistinctTagCount(int distinctTagCount)
```

---

---

Package

**tud.iir.classification.query**

# tud.iir.classification.query

## Class MapQuery

java.lang.Object

└--tud.iir.classification.query.MapQuery

---

public class **MapQuery**  
extends java.lang.Object

Map a query to an entity.

**Author:**

David

---

## Constructors

### MapQuery

public **MapQuery**()

## Methods

### main

public static void **main**(java.lang.String[] args)

**Parameters:**

args

## tud.iir.classification.query

### Class QueryWord

java.lang.Object

└--tud.iir.classification.query.QueryWord

---

```
public class QueryWord
extends java.lang.Object
```

---

## Fields

### LEFT

```
public static int LEFT
```

---

### RIGHT

```
public static int RIGHT
```

---

## Constructors

### QueryWord

```
public QueryWord(java.lang.String rootWord)
```

## Methods

### getRootWord

```
public java.lang.String getRootWord()
```

---

### setRootWord

```
public void setRootWord(java.lang.String rootWord)
```

---

### addWord

```
public void addWord(java.lang.String word,
                    int leftRight,
                    int position)
```

---

## getFullEntityName

```
public java.lang.String getFullEntityName()
```

Try to create a full entity name.

**Returns:**

---

Package

**tud.iir.classification.snippet**



# tud.iir.classification.snippet Class SnippetClassifier

java.lang.Object

└-[tud.iir.classification.Classifier](#)

└-**tud.iir.classification.snippet.SnippetClassifier**

---

public class **SnippetClassifier**  
extends [Classifier](#)

The SnippetClassifier is used to calculate prediction scores used for ranking of snippets according to their estimated quality. This class is described in detail in "Friedrich, Christopher. WebSnippets - Extracting and Ranking of entity-centric knowledge from the Web. Diploma thesis, Technische Universität Dresden, April 2010".

**Author:**

Christopher Friedrich

---

## Constructors

### SnippetClassifier

public **SnippetClassifier**()

## Methods

### classify

public float **classify**([Snippet](#) snippet)

Calculate the regression value for a given Snippet.

**Parameters:**

snippet - the snippet being regressed

**Returns:**

the regression value

---

### trainClassifier

public boolean **trainClassifier**(int conceptID,  
java.sql.PreparedStatement featureString,  
java.sql.PreparedStatement classificationString)

Train a classifier with the samples save in the database. The classifier is trained on a concept level.

**Parameters:**

conceptID - The id of the concept for which the classifier should be trained.

featureString - The SQL query string with the desired features to train the classifier.

---

(continued from last page)

## useTrainedClassifier

```
public void useTrainedClassifier()
```

Use an already trained classifier.

---

## main

```
public static void main(java.lang.String[] args)
```

### Parameters:

args

---

Package  
**tud.iir.control**

# tud.iir.control Class Controller

```
java.lang.Object
└--tud.iir.control.Controller
```

```
public class Controller
extends java.lang.Object
```

This class is the entry point to the WebKnox Core application.

**Author:**  
David Urbansky

## Fields

### NAME

```
public static final java.lang.String NAME
```

Constant value: **WebKnox**

### ID

```
public static final java.lang.String ID
```

Constant value: **WebKnox**

### VERSION

```
public static final double VERSION
```

Constant value: **0.12**

### WEB

```
public static final int WEB
```

Constant value: **1**

### SELECTION

```
public static final int SELECTION
```

Constant value: **2**

### SELECTION\_HALF

```
public static final int SELECTION_HALF
```

(continued from last page)

Constant value: 3

---

## EXTRACTION\_SOURCES

```
public static final int EXTRACTION_SOURCES
```

Constant value: 1

## Methods

### getInstance

```
public static Controller getInstance()
```

Get the instance of the class.

**Returns:**

---

### getConfig

```
public static PropertiesConfiguration getConfig()
```

---

### main

```
public static void main(java.lang.String[] args)  
    throws java.lang.Exception
```

WebKnox Core application entry point.

**Parameters:**

`args` - No arguments are read.

---

Package

**tud.iir.daterecognition**

# tud.iir.daterecognition

## Class DateConverter

java.lang.Object

└-- tud.iir.daterecognition.DateConverter

---

public class **DateConverter**  
extends java.lang.Object

---

### Fields

#### TECH\_URL

public static final int **TECH\_URL**

Constant value: 1

---

#### TECH\_HTTP\_HEADER

public static final int **TECH\_HTTP\_HEADER**

Constant value: 2

---

#### TECH\_HTML\_HEAD

public static final int **TECH\_HTML\_HEAD**

Constant value: 3

---

#### TECH\_HTML\_STRUC

public static final int **TECH\_HTML\_STRUC**

Constant value: 4

---

#### TECH\_HTML\_CONT

public static final int **TECH\_HTML\_CONT**

Constant value: 5

---

#### TECH\_REFERENCE

public static final int **TECH\_REFERENCE**

Constant value: 6

---

## TECH\_ARCHIVE

```
public static final int TECH_ARCHIVE
```

Constant value: 7

## Constructors

### DateConverter

```
public DateConverter()
```

## Methods

### convert

```
public static java.lang.Object convert(ExtractedDate date,  
    int techniqueFlag)
```



# tud.iir.daterecognition

## Class DateEvaluator

java.lang.Object

└--tud.iir.daterecognition.DateEvaluator

---

public class **DateEvaluator**  
extends java.lang.Object

---

## Constructors

### DateEvaluator

public **DateEvaluator**()

---

### DateEvaluator

public **DateEvaluator**(java.lang.String url)

---

### DateEvaluator

public **DateEvaluator**(java.lang.String url,  
boolean referenceLookUp)

---

## Methods

### evaluate

public java.util.HashMap **evaluate**(java.util.ArrayList extractedDates)

---

### checkDayMonthYearOrder

public void **checkDayMonthYearOrder**(java.lang.Object orginalDate,  
java.util.HashMap toCheckcDates)

#### See Also:

DateEvaluatorHelper.checkDayMonthYearOrder

---

### deployMetaDates

public static java.util.HashMap **deployMetaDates**(java.util.HashMap metaDates,  
java.util.HashMap dates)

---

(continued from last page)

---

## evaluateURLDate

```
public static java.util.HashMap evaluateURLDate(java.util.ArrayList dates)
```

Evaluates the URL dates.

### Parameters:

dates

### Returns:

---

## setUrl

```
public void setUrl(java.lang.String url)
```

---

## getUrl

```
public java.lang.String getUrl()
```

---

## setReferneceLookUp

```
public void setReferneceLookUp(boolean referneceLookUp)
```

# tud.iir.daterecognition

## Class DateEvaluatorHelper

java.lang.Object

└-- tud.iir.daterecognition.DateEvaluatorHelper

public class **DateEvaluatorHelper**  
extends java.lang.Object

## Constructors

### DateEvaluatorHelper

public **DateEvaluatorHelper**()

## Methods

### isDateInRange

public static boolean **isDateInRange**([ExtractedDate](#) date)

Checks if a date is between 13th of November 1990, time 0:00 and now.

**Parameters:**

date

**Returns:**

### getHighestRate

public static java.util.HashMap **getHighestRate**(java.util.HashMap dates)

Returns the date with highest rate.

**Parameters:**

dates

**Returns:**

Hashmap with a single entry.

### evaluateTag

public static java.util.HashMap **evaluateTag**(java.util.HashMap contentDates)

Increase the rate by 10 percent, if date surrounding tag is a headline-tag.

**Parameters:**

contentDates

(continued from last page)

**Returns:**

---

## evaluateKeyLocAttr

```
public static java.util.HashMap evaluateKeyLocAttr(java.util.ArrayList attrDates)
```

Calculates rate of dates with keyword within attribute.

**Parameters:**

attrDates

**Returns:**

---

## setRateWhightedByGroups

```
public static void setRateWhightedByGroups(java.util.ArrayList datesToSet,  
java.util.HashMap dates)
```

Calculates the rate for dates.

$\text{NewRate} = \text{CountOfSameDatesToSet} / \text{CountOfDatesToSet}$ .

Example: datesToSet.size()=5; 3/5 and 2/5.

**Parameters:**

datesToSet

dates

---

## setRateWhightedByGroups

```
public static void setRateWhightedByGroups(java.util.ArrayList datesToSet,  
java.util.HashMap dates,  
int stopFlag)
```

Calculates the rate for dates.

$\text{NewRate} = \text{CountOfSameDatesToSet} / \text{CountOfDatesToSet}$ .

Example: datesToSet.size()=5; 3/5 and 2/5.

**Parameters:**

datesToSet

dates

---

## setRateToZero

```
public static void setRateToZero(java.util.ArrayList datesToBeSetZero,  
java.util.HashMap map)
```

Sets for all dates from arraylist the rate-value to 0.0 in map.

**Parameters:**

datesToBeSetZero

map

---

## setRat

```
public static void setRat(java.util.ArrayList datesToBeSetZero,  
java.util.HashMap map,  
double rate)
```

(continued from last page)

Sets for all dates from arraylist the rate-value to given value in map.

**Parameters:**

datesToBeSetZero  
map

---

## evaluateKeyLocCont

```
public static java.util.HashMap evaluateKeyLocCont(java.util.ArrayList contDates)
```

Calculates the rate of dates with keywords within content.

**Parameters:**

contDates

**Returns:**


---

## checkDayMonthYearOrder

```
public static void checkDayMonthYearOrder(java.lang.Object orginalDate,  
ExtractedDate toCheckDate)
```

Compares a date1 with a well known date2, where you are sure that this is in the right format.

To make this sure, the format will be checked automatically. (Formats are `RegExp.DATE_URL_D`, `RegExp.DATE_URL_MMMM_D`, `RegExp.DATE_ISO8601_YMD` and `RegExp.DATE_ISO8601_YMD_NO`).

If date1 and date2 have equal years and day and month are mixed up, month and day in date2 will be exchanged.

Caution, no other parameters will be changed. So the original datestring and format will stay, and if you call `ExtractedDate.setDateParticles` old values will be rest.

Example: date1: 2010-09-07; date2: 07/09/2010, but will be identified as US-American-date to 2010-07-09.

date2 month and day will be exchanged so you get 2010-09-07 by calling

`ExtractedDate.getNormalizedDate`.

**Parameters:**

orginalDate  
toCheckDate

---

## getKeywordPriority

```
public static byte getKeywordPriority(ExtractedDate date)
```

Returns the classpriority of a keyword. If a date has no keyword -1 will be returned.

Otherwise returning values are equal to [KeyWords](#) static values.

**Parameters:**

date

**Returns:**


---

## calcContDateAttr

```
public static double calcContDateAttr(ContentDate date)
```

Sets the factor for rate-calculation of dates with keywords within attributes.

**Parameters:**

date

(continued from last page)

**Returns:**

---

## calcContDateContent

```
public static double calcContDateContent(ContentDate date)
```

Sets the factor for rate-calculation of dates with keywords within content.

**Parameters:**

date

**Returns:**

# tud.iir.daterecognition

## Class DateGetter

```
java.lang.Object
└─ tud.iir.daterecognition.DateGetter
```

```
public class DateGetter
extends java.lang.Object
```

DateGetter provides methods for getting dates from URL and rate them.

**Parameters:**

T

**Author:**

Martin Gregor (mail@m-gregor.de)

## Constructors

### DateGetter

```
public DateGetter()
```

### DateGetter

```
public DateGetter(java.lang.String url)
```

Constructor creates a new DateGetter with a given URL.

**Parameters:**

url - URL that will be analyzed

### DateGetter

```
public DateGetter(org.w3c.dom.Document document)
```

### DateGetter

```
public DateGetter(java.lang.String url,
                  org.w3c.dom.Document document)
```

## Methods

### getDate

```
public java.util.ArrayList getDate()
```

Analyzes a webpage by different techniques to find dates. The techniques are found in DateGetterHelper.

Type of the found dates are ExtractedDate.

(continued from last page)

**Returns:**

A array of ExtractedDates.

---

**getURL**

```
public java.lang.String getURL()
```

Getter for global variable URL.

**Returns:**

URL.

---

**setURL**

```
public void setURL(java.lang.String url)
```

Setter for global variable URL.

**Returns:**

URL.

---

**setTechHTTP**

```
public void setTechHTTP(boolean value)
```

---

**setTechURL**

```
public void setTechURL(boolean value)
```

---

**setTechHTMLHead**

```
public void setTechHTMLHead(boolean value)
```

---

**setTechHTMLStruct**

```
public void setTechHTMLStruct(boolean value)
```

---

**setTechHTMLContent**

```
public void setTechHTMLContent(boolean value)
```

---

**setTechReference**

```
public void setTechReference(boolean value)
```



---

## setTechArchive

```
public void setTechArchive(boolean value)
```

---

## setAllFalse

```
public void setAllFalse()
```

---

## setAllTrue

```
public void setAllTrue()
```

# tud.iir.daterecognition

## Class DateGetterHelper

```
java.lang.Object
└-- tud.iir.daterecognition.DateGetterHelper
```

---

```
public final class DateGetterHelper
extends java.lang.Object
```

DateGetterHelper provides the techniques to find dates out of webpages. Also provides different helper methods.

**Author:**  
Martin Gregor

---

## Methods

### getURLDate

```
public static URLDate getURLDate(java.lang.String url)
```

looks up for a date in the URL

**Parameters:**

url

**Returns:**

a extracted Date

---

### getHTTPHeaderDate

```
public static java.util.ArrayList getHTTPHeaderDate(java.lang.String url)
```

Extracts date form HTTP-header, that is written in "Last-Modified"-tag.

**Parameters:**

url

**Returns:**

The extracted Date.

---

### getStructureDate

```
public static java.util.ArrayList getStructureDate(org.w3c.dom.Document document)
```

---

### getBodyStructureDates

```
public static java.util.ArrayList getBodyStructureDates(org.w3c.dom.Document document)
```

---

(continued from last page)

---

## getChildrenDates

```
public static java.util.ArrayList getChildrenDates(org.w3c.dom.Node node,  
int depth)
```

---

## checkForDate

```
public static StructureDate checkForDate(org.w3c.dom.Node node)
```

Looks up in a [TAG](#) for [ATTRIBUTES](#) .

Trays to find dates in the attributes.

If a date is found, looks for a date-keywords in the other attributes.

If one is found, we got the context for the date, otherwise we use attribute-name for context.

The "href"-attribute will not be checked, because we will do this in "links-out-technique" with `getURLDate()`.

### Parameters:

node - to check

### Returns:

A `ExtractedDate` with Context.

---

## getHeadDates

```
public static java.util.ArrayList getHeadDates(org.w3c.dom.Document document)
```

Finds dates in head-part of a webpage.

### Parameters:

document

### Returns:

a array-list with dates.

---

## findDate

```
public static ExtractedDate findDate(java.lang.String dateString)
```

Tries to match a date in a dateformat. The format is given by the regular expressions of `RegExp`.

### Parameters:

dateString - a date to match.

### Returns:

The found format, defined in `RegExp` constants.

If no match is found return `null`.

---

## findDate

```
public static ExtractedDate findDate(java.lang.String dateString,  
java.lang.Object[] regExpArray)
```

Tries to match a date in a dateformat. The format is given by the regular expressions of `RegExp`.

### Parameters:

dateString - a date to match.

regExpArray - regular expressions of dates to match. If this is null [RegExp](#).getAllRegExp will be called.

---

(continued from last page)

**Returns:**

The found format, defined in RegExp constants.  
If no match is found return **null**.

---

## findALLDates

```
public static java.util.ArrayList findALLDates(java.lang.String text)
```

**Parameters:**

dateString - a date to match.

**Returns:**

The found format, defined in RegExp constants.  
If no match is found return **null**.

---

## getWhitespaces

```
public static java.lang.String getWhitespaces(java.lang.String text)
```

## hasKeyword

```
public static java.lang.String hasKeyword(java.lang.String text,  
java.lang.String[] keys)
```

Check a string for keywords. Used to look in tag-values for date-keys.

**Parameters:**

text - string with possible keywords.  
keys - a array of keywords.

**Returns:**

the found keyword.

---

## getSeparator

```
public static java.lang.String getSeparator(ExtractedDate date)
```

Finds out the separating symbol of date-string

**Parameters:**

date

**Returns:**

## getDateFromString

```
public static ExtractedDate getDateFromString(java.lang.String dateString,  
java.lang.String[] regExp)
```

**Parameters:**

string - string, which is to be searched

---

(continued from last page)

regExp - regular expression for search

offsetStart - is slider for beginning substring (no negative values) - e.g. substring: "abcd"  
offsetStart=0: "abcd" offsetStart=1: "bcd" offsetStart=-1: "abcd"

**Returns:**

found substring or null

---

## getContentDates

```
public static java.util.ArrayList getContentDates(org.w3c.dom.Document document)
```

---

## enterTextnodes

```
public static java.util.ArrayList enterTextnodes(org.w3c.dom.Node node,  
    java.lang.String doc,  
    int depth)
```

---

## checkTextNode

```
public static java.util.ArrayList checkTextNode(org.w3c.dom.Text node,  
    java.lang.String doc,  
    int depth)
```

---

## findNodeKeywordPart

```
public static java.lang.String findNodeKeywordPart(org.w3c.dom.Node node,  
    java.lang.String[] keyWords)
```

---

## findNodeKeyword

```
public static java.lang.String findNodeKeyword(org.w3c.dom.Node node,  
    java.lang.String[] keyWords)
```

---

## setNearestTextkeyword

```
public static ContentDate setNearestTextkeyword(java.lang.String textString,  
    ContentDate date)
```

---

## getReferenceDates

```
public static java.util.ArrayList getReferenceDates(org.w3c.dom.Document document)
```

---

(continued from last page)

## getReferenceDates

```
public static java.util.ArrayList getReferenceDates(org.w3c.dom.Document document,  
int maxLinks)
```

# tud.iir.daterecognition

## Class DateGetterMain

java.lang.Object

└─ tud.iir.daterecognition.DateGetterMain

---

public class **DateGetterMain**  
extends java.lang.Object

---

## Constructors

### DateGetterMain

public **DateGetterMain**()

## Methods

### main

public static void **main**(java.lang.String[] args)

#### Parameters:

args

# tud.iir.daterecognition

## Class ExtractedDateHelper

```
java.lang.Object
└-- tud.iir.daterecognition.ExtractedDateHelper
```

---

```
public class ExtractedDateHelper
extends java.lang.Object
```

---

## Constructors

### ExtractedDateHelper

```
public ExtractedDateHelper()
```

## Methods

### getMonthNumber

```
public static java.lang.String getMonthNumber(java.lang.String monthString)
```

convert month-name in a number; January is 01..

**Parameters:**

month

**Returns:**

month-number as string

---

### normalizeYear

```
public static int normalizeYear(java.lang.String year)
```

Normalizes a year. Removes apostrophe (e.g. '99) and makes it four digit.

**Parameters:**

year

**Returns:**

A four digit year.

---

### removeNodigits

```
public static java.lang.String removeNodigits(java.lang.String datePart)
```

Removes the symbols "'" from Year '99 and "," from Day 03, June.

**Parameters:**

date

---



(continued from last page)

**Returns:**the entered date without the symbols

---

## get4DigitYear

```
public static int get4DigitYear(int year)
```

Sets the year in 4 digits format.

E.g.: year = 12; current year = 2010 -&gt; year &gt; 10 -&gt; 1912

year = 7; current year = 2010 -&gt; year &lt; 10 -&gt; 2007

year = 10; current year = 2010 -&gt; year &gt; 10 -&gt; 2010

year = 99; current year = 2010 -&gt; year &gt; 10 -&gt; 1999

**Parameters:**

date

**Returns:**

---

## getSeparator

```
public static java.lang.String getSeparator(java.lang.String text)
```

**Parameters:**

text - a date, where year, month and day are separated by . / or \_

**Returns:**the separating symbol

---

## get2Digits

```
public static java.lang.String get2Digits(int number)
```

Adds a leading zero for numbers less then ten.

E.g.: 3 -&gt; "03"; 12 -&gt; "12"; 386 -&gt; "376" ...

**Parameters:**

number

**Returns:**a minimum two digit number

---

## createActualDate

```
public static ExtractedDate createActualDate()
```

Crates a extracted date with actual date and time in UTC timezone.

Thereby format YYYY-MM-DDTHH:MM:SSZ is used.

**Returns:**Extracted date.

---

## createActualDate

```
public static ExtractedDate createActualDate(java.util.Locale local)
```

---

---

## removeTimezone

```
public static java.lang.String[] removeTimezone(java.lang.String dateString)
```

Removes timezone acronyms.

**Parameters:**

dateString

**Returns:**

---

## getTypString

```
public static java.lang.String getTypString(int typ)
```

Returns a extracted date type in a human readable string.

**Parameters:**

typ

**Returns:**

## tud.iir.daterecognition Class LinkSetCreator

java.lang.Object

└─ tud.iir.daterecognition.LinkSetCreator

---

public class **LinkSetCreator**  
extends java.lang.Object

---

### Constructors

#### LinkSetCreator

public **LinkSetCreator**()

### Methods

#### main

public static void **main**(java.lang.String[] args)

# tud.iir.daterecognition

## Class testCrawler

java.lang.Object

└--tud.iir.daterecognition.testCrawler

---

```
public class testCrawler
extends java.lang.Object
```

---

## Fields

### countSame

```
public static java.lang.Integer countSame
```

---

### countAll

```
public static java.lang.Integer countAll
```

---

### countThreads

```
public static java.lang.Integer countThreads
```

---

### file

```
public static final java.io.File file
```

---

## Constructors

### testCrawler

```
public testCrawler()
```

---

## Methods

### main

```
public static void main(java.lang.String[] args)
```

**Parameters:**

---

(continued from last page)

args

---

## checkURLs

```
public static void checkURLs()
```

---

## checkLinkSet

```
public static void checkLinkSet()
```

---

## crawlURLwithDate

```
public static void crawlURLwithDate()
```

---

## evaluateURLwithDate

```
public static void evaluateURLwithDate()
```

---

## addStats

```
public static void addStats(ExtractedDate url,  
    double urlRate,  
    java.util.Map.Entry otherDate,  
    double highestRate,  
    boolean otherIsHighestDate)
```

---

Package

**tud.iir.daterecognition.dates**

# tud.iir.daterecognition.dates

## Class BodyDate

```
java.lang.Object
├── tud.iir.daterecognition.dates.ExtractedDate
│   ├── tud.iir.daterecognition.dates.KeywordDate
│   │   └── tud.iir.daterecognition.dates.BodyDate
```

Direct Known Subclasses:

[ContentDate](#), [StructureDate](#)

---

```
public abstract class BodyDate
extends KeywordDate
```

---

## Fields

### STRUCTURE\_DEPTH

```
public static final int STRUCTURE_DEPTH
```

Constant value: 101

## Constructors

### BodyDate

```
public BodyDate()
```

---

### BodyDate

```
public BodyDate(java.lang.String dateString)
```

Parameters:

dateString

---

### BodyDate

```
public BodyDate(java.lang.String dateString,
                java.lang.String format)
```

Parameters:

dateString

format

(continued from last page)

## Methods

### setTag

```
public void setTag(java.lang.String tag)
```

---

### getTag

```
public java.lang.String getTag()
```

---

### get

```
public int get(int field)
```

---

### set

```
public void set(int field,  
                int value)
```

---

### toString

```
public java.lang.String toString()
```

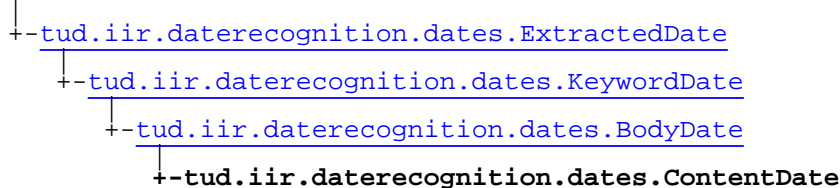
---



# tud.iir.daterecognition.dates

## Class ContentDate

java.lang.Object



public class **ContentDate**  
extends [BodyDate](#)

**Author:**  
Martin Gregor

## Fields

### KEY\_LOC\_ATTR

public static final int **KEY\_LOC\_ATTR**

Keyword found in attribute of surrounding tag.  
Constant value: 201

### KEY\_LOC\_CONTENT

public static final int **KEY\_LOC\_CONTENT**

Keyword found in text (content) of surrounding tag.  
Constant value: 202

### DATEPOS\_IN\_TAGTEXT

public static final int **DATEPOS\_IN\_TAGTEXT**

Position of datestring in text of found tag.  
Constant value: 201

### DISTANCE\_DATE\_KEYWORD

public static final int **DISTANCE\_DATE\_KEYWORD**

Distance between datestring and nearest found keyword.  
Constant value: 202

### KEYWORDLOCATION

public static final int **KEYWORDLOCATION**

Location of keyword. In tagtext (content), attribute or tagname.  
Constant value: 203

(continued from last page)

## DATEPOS\_IN\_DOC

```
public static final int DATEPOS_IN_DOC
```

Position of datestring in text of whole document.  
Constant value: 204

## Constructors

### ContentDate

```
public ContentDate()
```

---

### ContentDate

```
public ContentDate(java.lang.String dateString)
```

#### Parameters:

dateString

---

### ContentDate

```
public ContentDate(java.lang.String dateString,  
                  java.lang.String format)
```

#### Parameters:

dateString

format

## Methods

### getType

```
public int getType()
```

---

### getKeyLocToString

```
public java.lang.String getKeyLocToString()
```

---

### toString

```
public java.lang.String toString()
```

---

### get

```
public int get(int field)
```

(continued from last page)

---

## set

```
public void set(int field,  
               int value)
```

# tud.iir.daterecognition.dates

## Class ExtractedDate

java.lang.Object

└-- tud.iir.daterecognition.dates.ExtractedDate

Direct Known Subclasses:

[KeywordDate](#), [ReferenceDate](#), [URLDate](#)

---

```
public class ExtractedDate
extends java.lang.Object
```

Represents a date, found in a webpage.

A object will be created with a date-string and a possible format.

It can be asked for year, month, day and time. If some values can not be constructed the value will be -1.

Author:

Martin Gregor\*

---

## Fields

### TECH\_URL

```
public static final int TECH_URL
```

Constant value: 1

---

### TECH\_HTTP\_HEADER

```
public static final int TECH_HTTP_HEADER
```

Constant value: 2

---

### TECH\_HTML\_HEAD

```
public static final int TECH_HTML_HEAD
```

Constant value: 3

---

### TECH\_HTML\_STRUC

```
public static final int TECH_HTML_STRUC
```

Constant value: 4

---

### TECH\_HTML\_CONT

```
public static final int TECH_HTML_CONT
```

Constant value: 5

---

## TECH\_REFERENCE

```
public static final int TECH_REFERENCE
```

Constant value: 6

---

## TECH\_ARCHIVE

```
public static final int TECH_ARCHIVE
```

Constant value: 7

---

## YEAR

```
public static final int YEAR
```

Constant value: 1

---

## MONTH

```
public static final int MONTH
```

Constant value: 2

---

## DAY

```
public static final int DAY
```

Constant value: 3

---

## HOURL

```
public static final int HOURL
```

Constant value: 4

---

## MINUTE

```
public static final int MINUTE
```

Constant value: 5

---

## SECOND

```
public static final int SECOND
```

Constant value: 6

## Constructors

(continued from last page)

---

## ExtractedDate

```
public ExtractedDate()
```

Standard constructor.

---

## ExtractedDate

```
public ExtractedDate(java.lang.String dateString)
```

Creates a new date and sets the dateString.

**Parameters:**

dateString

---

## ExtractedDate

```
public ExtractedDate(java.lang.String dateString,  
                     java.lang.String format)
```

creates a new date and sets dateString and format

**Parameters:**

dateString

format

---

## Methods

### getNormalizedDate

```
public java.lang.String getNormalizedDate()
```

Constructs a normalized datestring in a format from YYYY-MM-DD HH:MM:SS to YYYY-MM depending of given values

**Parameters:**

dateParts

**Returns:**

---

### setDateString

```
public void setDateString(java.lang.String dateString)
```

**Parameters:**

dateString

---

### getDateString

```
public java.lang.String getDateString()
```

**Returns:**

---

---

## setFormat

```
public void setFormat(java.lang.String format)
```

---

---

## getFormat

```
public java.lang.String getFormat()
```

---

---

## get

```
public int get(int field)
```

---

---

## getAll

```
public java.util.ArrayList getAll()
```

---

---

## setAll

```
public void setAll(java.util.ArrayList values)
```

---

---

## set

```
public void set(int field,  
               int value)
```

---

---

## toString

```
public java.lang.String toString()
```

---

---

## getType

```
public int getType()
```

---

---

## setUrl

```
public void setUrl(java.lang.String url)
```

---

(continued from last page)

## getUrl

```
public java.lang.String getUrl()
```

---

## getExactness

```
public int getExactness()
```

---

## getKeyword

```
public java.lang.String getKeyword()
```



# tud.iir.daterecognition.dates

## Class HeadDate

```
java.lang.Object
├── tud.iir.daterecognition.dates.ExtractedDate
│   ├── tud.iir.daterecognition.dates.KeywordDate
│   │   └── tud.iir.daterecognition.dates.HeadDate
```

```
public class HeadDate
extends KeywordDate
```

Author:  
salco

## Constructors

### HeadDate

```
public HeadDate()
```

### HeadDate

```
public HeadDate(java.lang.String dateString)
```

Parameters:  
dateString

### HeadDate

```
public HeadDate(java.lang.String dateString,
                java.lang.String format)
```

Parameters:  
dateString  
format

## Methods

### getType

```
public int getType()
```

(continued from last page)

## getTag

```
public java.lang.String getTag()
```

---

## setTag

```
public void setTag(java.lang.String tag)
```

# tud.iir.daterecognition.dates

## Class HTTPDate

```
java.lang.Object
├── tud.iir.daterecognition.dates.ExtractedDate
│   ├── tud.iir.daterecognition.dates.KeywordDate
│   └── tud.iir.daterecognition.dates.HTTPDate
```

---

```
public class HTTPDate
extends KeywordDate
```

Author:  
salco

---

## Constructors

### HTTPDate

```
public HTTPDate()
```

---

### HTTPDate

```
public HTTPDate(java.lang.String dateString)
```

Parameters:  
dateString

---

### HTTPDate

```
public HTTPDate(java.lang.String dateString,
                java.lang.String format)
```

Parameters:  
dateString  
format

---

## Methods

### getType

```
public int getType()
```

# tud.iir.daterecognition.dates

## Class KeywordDate

```
java.lang.Object
├── tud.iir.daterecognition.dates.ExtractedDate
│   └── tud.iir.daterecognition.dates.KeywordDate
```

Direct Known Subclasses:

[BodyDate](#), [HeadDate](#), [HTTPDate](#)

---

```
public abstract class KeywordDate
extends ExtractedDate
```

---

## Constructors

### KeywordDate

```
public KeywordDate()
```

---

### KeywordDate

```
public KeywordDate(java.lang.String dateString)
```

Parameters:

dateString

---

### KeywordDate

```
public KeywordDate(java.lang.String dateString,
                    java.lang.String format)
```

Parameters:

dateString

format

---

## Methods

### toString

```
public java.lang.String toString()
```

---

(continued from last page)

## setKeyword

```
public void setKeyword(java.lang.String keyword)
```

---

## getKeyword

```
public java.lang.String getKeyword()
```

# tud.iir.daterecognition.dates

## Class ReferenceDate

```
java.lang.Object
├── tud.iir.daterecognition.dates.ExtractedDate
│   └── tud.iir.daterecognition.dates.ReferenceDate
```

```
public class ReferenceDate
extends ExtractedDate
```

## Fields

### RATE

```
public static final int RATE
```

Constant value: 101

## Constructors

### ReferenceDate

```
public ReferenceDate()
```

### ReferenceDate

```
public ReferenceDate(java.lang.String dateString)
```

### ReferenceDate

```
public ReferenceDate(java.lang.String dateString,
                     java.lang.String format)
```

## Methods

### get

```
public int get(int field)
```

(continued from last page)

## set

```
public void set(int field,  
               int value)
```

---

## getType

```
public int getType()
```

# tud.iir.daterecognition.dates Class StructureDate

```
java.lang.Object
├── tud.iir.daterecognition.dates.ExtractedDate
│   ├── tud.iir.daterecognition.dates.KeywordDate
│   │   ├── tud.iir.daterecognition.dates.BodyDate
│   │   │   └── tud.iir.daterecognition.dates.StructureDate
```

```
public class StructureDate
extends BodyDate
```

Author:  
salco

## Constructors

### StructureDate

```
public StructureDate()
```

### StructureDate

```
public StructureDate(java.lang.String dateString)
```

Parameters:  
dateString

### StructureDate

```
public StructureDate(java.lang.String dateString,
                     java.lang.String format)
```

Parameters:  
dateString  
format

## Methods

### getType

```
public int getType()
```



# tud.iir.daterecognition.dates

## Class URLDate

```
java.lang.Object
├── tud.iir.daterecognition.dates.ExtractedDate
│   └── tud.iir.daterecognition.dates.URLDate
```

---

```
public class URLDate
extends ExtractedDate
```

---

## Constructors

### URLDate

```
public URLDate()
```

---

### URLDate

```
public URLDate(java.lang.String dateString)
```

---

### URLDate

```
public URLDate(java.lang.String dateString,
               java.lang.String format)
```

---

## Methods

### getType

```
public int getType()
```

---

### toString

```
public java.lang.String toString()
```

---

---

Package  
**tud.iir.extraction**

# tud.iir.extraction

## Class ConceptDateComparator

```
java.lang.Object
└--tud.iir.extraction.ConceptDateComparator
```

### All Implemented Interfaces:

java.io.Serializable, java.util.Comparator

---

```
public class ConceptDateComparator
    extends java.lang.Object
    implements java.util.Comparator, java.io.Serializable
```

Sort concepts by the date they were last searched.

### Author:

David Urbansky

---

## Constructors

### ConceptDateComparator

```
public ConceptDateComparator()
```

## Methods

### compare

```
public int compare(Concept c1,
                  Concept c2)
```

Oldest concept first (null before that as null means "never searched so far").

### Parameters:

- c1 - Concept1
- c2 - Concept2

## tud.iir.extraction

# Class ExtractionProcessManager

```
java.lang.Object
└--tud.iir.extraction.ExtractionProcessManager
```

---

```
public class ExtractionProcessManager
extends java.lang.Object
```

The ExtractionProcessManager manages the entity and the fact extraction process.

**Author:**  
David Urbansky

---

## Fields

### entityExtractionIsRunning

```
public static boolean entityExtractionIsRunning
```

---

### factExtractionIsRunning

```
public static boolean factExtractionIsRunning
```

---

### qaExtractionIsRunning

```
public static boolean qaExtractionIsRunning
```

---

### snippetExtractionIsRunning

```
public static boolean snippetExtractionIsRunning
```

---

### mioExtractionIsRunning

```
public static boolean mioExtractionIsRunning
```

---

### QUANTITY\_TRUST

```
public static final int QUANTITY_TRUST
```

Constant value: 1

---

(continued from last page)

---

## SOURCE\_TRUST

```
public static final int SOURCE_TRUST
```

Constant value: 2

---

## EXTRACTION\_TYPE\_TRUST

```
public static final int EXTRACTION_TYPE_TRUST
```

Constant value: 3

---

## COMBINED\_TRUST

```
public static final int COMBINED_TRUST
```

Constant value: 4

---

## CROSS\_TRUST

```
public static final int CROSS_TRUST
```

Constant value: 5

---

## BENCHMARK\_FULL\_SET

```
public static final int BENCHMARK_FULL_SET
```

Constant value: 1

---

## BENCHMARK\_HALF\_SET

```
public static final int BENCHMARK_HALF_SET
```

Constant value: 2

---

## MICROSOFT\_8

```
public static final int MICROSOFT_8
```

Constant value: 1

---

## YAHOO\_8

```
public static final int YAHOO_8
```

Constant value: 2

---

## HAKIA\_8

```
public static final int HAKIA_8
```

---

(continued from last page)

Constant value: 3

---

## GOOGLE\_8

```
public static final int GOOGLE_8
```

Constant value: 4

---

## BENCHMARK\_FACT\_EXTRACTION

```
public static java.lang.String BENCHMARK_FACT_EXTRACTION
```

---

## BENCHMARK\_ENTITY\_EXTRACTION

```
public static java.lang.String BENCHMARK_ENTITY_EXTRACTION
```

---

## Constructors

### ExtractionProcessManager

```
public ExtractionProcessManager()
```

---

## Methods

### startEntityExtraction

```
public static void startEntityExtraction()
```

---

### stopEntityExtraction

```
public static boolean stopEntityExtraction()
```

---

### startFactExtraction

```
public static void startFactExtraction()
```

---

### stopFactExtraction

```
public static boolean stopFactExtraction()
```

---

(continued from last page)

---

## runFactExtractionBenchmark

```
public static void runFactExtractionBenchmark()
```

---

## startQAExtraction

```
public static void startQAExtraction()
```

---

## stopQAExtraction

```
public static boolean stopQAExtraction()
```

---

## startSnippetExtraction

```
public static void startSnippetExtraction()
```

---

## stopSnippetExtraction

```
public static boolean stopSnippetExtraction()
```

---

## startMIOExtraction

```
public static void startMIOExtraction()
```

---

## stopMIOExtraction

```
public static boolean stopMIOExtraction()
```

---

## startFullExtractionLoop

```
public static void startFullExtractionLoop()
```

---

## getSourceRetrievalSite

```
public static int getSourceRetrievalSite()
```

---

## getSourceRetrievalCount

```
public static int getSourceRetrievalCount()
```

---

(continued from last page)

---

## isUseConceptSynonyms

```
public static boolean isUseConceptSynonyms()
```

---

## setUseConceptSynonyms

```
public static void setUseConceptSynonyms(boolean useConceptSynonyms)
```

---

## isUseAttributeSynonyms

```
public static boolean isUseAttributeSynonyms()
```

---

## setUseAttributeSynonyms

```
public static void setUseAttributeSynonyms(boolean useAttributeSynonyms)
```

---

## isFindNewAttributesAndValues

```
public static boolean isFindNewAttributesAndValues()
```

---

## setFindNewAttributesAndValues

```
public static void setFindNewAttributesAndValues(boolean findNewAttributesAndValues)
```

---

## isContinueQAExtraction

```
public static boolean isContinueQAExtraction()
```

---

## setContinueQAExtraction

```
public static void setContinueQAExtraction(boolean continueQAExtraction)
```

---

## getBenchmarkSetSize

```
public static int getBenchmarkSetSize()
```

---



(continued from last page)

## setBenchmarkSetSize

```
public static void setBenchmarkSetSize(int benchmarkSetSize)
```

---

## getBenchmarkSet

```
public static int getBenchmarkSet()
```

---

## setBenchmarkSet

```
public static void setBenchmarkSet(int benchmarkSet)
```

---

## getBenchmarkType

```
public static java.lang.String getBenchmarkType()
```

---

## setBenchmarkType

```
public static void setBenchmarkType(java.lang.String benchmarkType)
```

---

## getTrustFormula

```
public static int getTrustFormula()
```

---

## setTrustFormula

```
public static void setTrustFormula(int trustFormula)
```

---

# tud.iir.extraction

## Class ExtractionType

java.lang.Object

└-- tud.iir.extraction.ExtractionType

---

public final class **ExtractionType**  
extends java.lang.Object

In the ExtractionType class the different extraction types are defined. Also the trust for each extraction type can be calculated.

**Author:**

David Urbansky

---

## Fields

### UNKNOWN

public static final int **UNKNOWN**

Constant value: 0

---

### USER\_INPUT

public static final int **USER\_INPUT**

Constant value: 15

---

### FREE\_TEXT\_SENTENCE

public static final int **FREE\_TEXT\_SENTENCE**

Constant value: 1

---

### STRUCTURED\_PHRASE

public static final int **STRUCTURED\_PHRASE**

Constant value: 2

---

### TABLE\_CELL

public static final int **TABLE\_CELL**

Constant value: 3

---

### PATTERN\_PHRASE

public static final int **PATTERN\_PHRASE**

(continued from last page)

Constant value: 4

---

## COLON\_PHRASE

```
public static final int COLON_PHRASE
```

Constant value: 5

---

## IMAGE

```
public static final int IMAGE
```

Constant value: 6

---

## ENTITY\_PHRASE

```
public static final int ENTITY_PHRASE
```

Constant value: 7

---

## ENTITY\_FOCUSED\_CRAWL

```
public static final int ENTITY_FOCUSED_CRAWL
```

Constant value: 8

---

## ENTITY\_SEED

```
public static final int ENTITY_SEED
```

Constant value: 9

---

## initialTrust

```
public static double initialTrust
```

---

## Constructors

### ExtractionType

```
public ExtractionType()
```

---

## Methods

### getTrust

```
public static double getTrust(int extractionType)
```

(continued from last page)

Every extraction type has a trust between 0 and 1 (which is the precision of the extraction type).

**Parameters:**

`extractionType` - The extraction type constant.

**Returns:**

The trust for the given extraction type.

---

## getTrust

```
public static double getTrust(int extractionType,  
    java.lang.String type)
```

Get the extraction type trust by type (concept, attribute or data type).

**Parameters:**

`extractionType` - The extraction type constant.  
`type` - A string that specifies the type.

**Returns:**

The trust for the given extraction type.

---

## addExtraction

```
public static void addExtraction(int extractionType,  
    boolean correct)
```

---

## addExtractionByType

```
public static void addExtractionByType(int extractionType,  
    java.lang.String type,  
    boolean correct)
```

# tud.iir.extraction

## Class Extractor

java.lang.Object

└─ tud.iir.extraction.Extractor

Direct Known Subclasses:

[SnippetExtractor](#), [QAExtractor](#), [MIOExtractor](#), [FactExtractor](#), [EventExtractor](#), [EntityExtractor](#)

public abstract class **Extractor**  
extends java.lang.Object

The abstract Extractor from which other singleton extractors inherit.

**Author:**

David Urbansky

## Fields

### URL\_BINARY\_BLACKLIST

public static final java.lang.String **URL\_BINARY\_BLACKLIST**

List of binary file extensions.

### URL\_TEXTUAL\_BLACKLIST

public static final java.lang.String **URL\_TEXTUAL\_BLACKLIST**

List of textual file extensions.

## Constructors

### Extractor

public **Extractor**()

## Methods

### getKnowledgeManager

public [KnowledgeManager](#) **getKnowledgeManager**()

### setKnowledgeManager

public void **setKnowledgeManager**([KnowledgeManager](#) knowledgeManager)

(continued from last page)

---

## getThreadCount

```
public int getThreadCount()
```

---

## increaseThreadCount

```
public void increaseThreadCount()
```

---

## decreaseThreadCount

```
public void decreaseThreadCount()
```

---

## isStopped

```
public boolean isStopped()
```

---

## setStopped

```
public void setStopped(boolean stopped)
```

---

## isBenchmark

```
public boolean isBenchmark()
```

---

## filterURLs

```
public java.util.List filterURLs(java.util.List urls)
```

Returns for a given list of URLs these which are not blacklisted (be sure to set a blacklist first)

---

## stopExtraction

```
public boolean stopExtraction(boolean saveResults)
```

---

## setBenchmark

```
public void setBenchmark(boolean benchmark)
```

---

## getLogger

```
public Logger getLogger()
```

---

(continued from last page)

---

## getBlackList

```
public java.util.Set getBlackList()
```

---

## addSuffixesToBlackList

```
public void addSuffixesToBlackList(java.lang.String[] nBlackList)
```

Allows to define the SuffixBlackList

## tud.iir.extraction Class Filter

```
java.lang.Object
└-- tud.iir.extraction.Filter
```

---

```
public class Filter
extends java.lang.Object
```

The Filter class specifies thresholds for entity and fact trusts.

**Author:**  
David Urbansky

---

### Fields

#### minEntityCorroboration

```
public static double minEntityCorroboration
```

#### minFactCorroboration

```
public static double minFactCorroboration
```

### Methods

#### getInstance

```
public static Filter getInstance()
```



# tud.iir.extraction

## Class PageAnalyzer

java.lang.Object

└--tud.iir.extraction.PageAnalyzer

---

public class **PageAnalyzer**  
extends java.lang.Object

The PageAnalyzer's responsibility is it to perform generic tasks on the DOM tree.

**Author:**

David Urbansky

---

## Constructors

### PageAnalyzer

public **PageAnalyzer**()

## Methods

### setDocument

public void **setDocument**(org.w3c.dom.Document document)

---

### setDocument

public void **setDocument**(java.lang.String url)

---

### getTitle

public java.lang.String **getTitle**()

tag of the web page. Find and return the content of the

**Returns:**

The title of the web page.

---

### getDocumentAsString

public java.lang.String **getDocumentAsString**()

---

### getDocumentTextDump

public java.lang.String **getDocumentTextDump**()

---

---

## getDocumentTextDump

```
public static java.lang.String getDocumentTextDump(org.w3c.dom.Document document)
```

---

## detectFactTable

```
public java.lang.String[] detectFactTable()
```

Try to find a table with at least 4 facts.

**Returns:**

A string array with 0: the xpath to the table row, 1: the first td index and 2: the number of rows.

---

## constructAllXPath

```
public java.util.LinkedHashSet constructAllXPath(java.lang.String keyword)
```

Get all xPaths to the specified keyword in the specified document. The function does not return duplicates.

**Parameters:**

document - The document.  
keyword - The keyword.

**Returns:**

---

## constructAllXPath

```
public java.util.LinkedHashSet constructAllXPath(org.w3c.dom.Document document,  
java.lang.String keyword)
```

---

## constructAllXPath

```
public java.util.LinkedHashSet constructAllXPath(java.lang.String keyword,  
boolean deleteAllIndices,  
boolean wordMatch)
```

---

## constructAllXPath

```
public java.util.LinkedHashSet constructAllXPath(org.w3c.dom.Document document,  
java.lang.String keyword,  
boolean deleteAllIndices,  
boolean wordMatch)
```

---

(continued from last page)

---

## keepXPathPointingTo

```
public static java.util.LinkedHashSet keepXPathPointingTo( java.util.LinkedHashSet
xPaths,
    java.lang.String[] targetNodes)
```

Keep only xPaths that point to one of the specified elements. For example: [/HTML, /HTML/BODY/P] and [P] => [/HTML/BODY/P]

**Parameters:**

xPaths  
targetNodes

**Returns:**

A set of xPaths that all point to one of the specified elements.

---

## makeMutualXPath

```
public java.lang.String makeMutualXPath(java.util.HashSet xPathSet)
```

Find a single XPath that is generalized and works for many xPaths from the xPathSet. If several generalized xPaths are found, take the one with the highest count.

**Parameters:**

xPathSet - A set of xPaths.

**Returns:**

A string representing the mutual XPath.

---

## constructXPath

```
public java.lang.String constructXPath(org.w3c.dom.Node node)
```

Construct a simple XPath from the root to the specified node.

**Parameters:**

node - The start node.

**Returns:**

The string of the constructed XPath.

---

## nodeInTable

```
public boolean nodeInTable(java.lang.String xPath,
    int lookBack)
```

Find out whether the node specified by the XPath is in a table (in a td cell).

**Parameters:**

xPath - The xpath string pointing to the node.  
lookBack - How many parent nodes should be taken into account, e.g. with a lookBack of 3 the xpath /div/table/tr/td/div/span/a/b is not considered in a table because there is too much structure in the cell (more than 3 parents of the last node are not table structures).

**Returns:**

True if given xpath points to a node in a table, else false.

---

(continued from last page)

## getCellPath

```
public java.lang.String getCellPath(java.lang.String xPath)
```

Get the xPath to the table cell where the given xPath is pointing to. e.g. /div/p/table/tr/td/a[5]/b => /div/p/table/tr/td

**Parameters:**

xPath - The xPath.

**Returns:**

The string representation of an xPath.

---

## getTargetNode

```
public java.lang.String getTargetNode(java.lang.String xpath)
```

Get the name of the node the given xPath is pointing to. e.g. /html/body/div/table[5]/tr/td[3]/p/a[4] => a

**Parameters:**

xpath - The xPath.

**Returns:**

The string representation of an xPath.

---

## nodeInBox

```
public boolean nodeInBox(java.lang.String xPath,  
    int lookBack)
```

Check whether a node is in a box. A box is the "p" and the "div" tag.

**Parameters:**

xPath - The xPath.

lookBack - How many parent nodes should be considered.

**Returns:**

True if the specified xPath is in a box, else false.

---

## findLastBoxSection

```
public java.lang.String findLastBoxSection(java.lang.String xPath)
```

Find the last box section ("p", "div", "td" or "th") of the given xPath. This is helpful as a certain term might be in a too deep structure and searched elements are around it. e.g. /table/tr/td/div[4]/span/b/a => /table/tr/td/div[4]

**Parameters:**

xPath - The xPath.

**Returns:**

The potentially shortened xPath if found, else the input xPath.

---

## getNextSibling

```
public java.lang.String getNextSibling(java.lang.String xPath)
```

## getNextSibling

```
public java.lang.String getNextSibling(java.lang.String xPath,
    boolean tableCellSibling)
```

Create an xpath that points to the next sibling of the node specified by the given xPath. e.g.  
/div/p/table[4]/tr[6]/td[1] => /div/p/table[4]/tr[6]/td[2] /div/p/table[4]/tr[6]/td[1]/div[4] =>  
/div/p/table[4]/tr[6]/td[1]/div[5] /div/p/table[4]/tr[6]/th/b/a => /div/p/table[4]/tr[6]/td[1]/b/a  
/div/p/table[4]/tr[6]/td => /div/p/table[4]/tr[7]/td ----- with tableCellSibling = true -----  
/div/p/table[4]/tr[6]/td[1]/div[4] => /div/p/table[4]/tr[6]/td[2]/div[4] (compare with above)  
/div/p/table[4]/tr[6]/th/div[4] => /div/p/table[4]/tr[6]/td[1]/div[4] TODO sometimes a spacer cell is  
between attribute and value: [http://www.smartone-vodafone.com/jsp/phone/english/detail\\_v3.jsp?id=662](http://www.smartone-vodafone.com/jsp/phone/english/detail_v3.jsp?id=662)

### Parameters:

xPath - The xPath

tableCellSibling - If true, only siblings of table cells (td,th) are searched.

### Returns:

The xpath pointing to the sibling.

---

## getNextTableCell

```
public java.lang.String getNextTableCell(java.lang.String xPath)
```

## getFirstTableCell

```
public java.lang.String getFirstTableCell(java.lang.String xPath)
```

Point xPath to first table cell. For example: //TABLE/TR/TD => //TABLE/TR/TD[1] //TABLE/TR/TD[1]  
=> //TABLE/TR/TD[1] //TABLE/TR/TH => //TABLE/TR/TH

### Parameters:

xPath - The xPath.

### Returns:

The xPath pointing to the first table cell of the deepest table.

---

## getNumberOfTableRows

```
public int getNumberOfTableRows(java.lang.String attributeXPath)
```

Get number of table rows.

### Parameters:

attributeXPath - This path should point to one attribute cell.

### Returns:

The number of table rows.

---

## getTableRows

```
public java.util.ArrayList getTableRows(java.lang.String attributeXPath)
```

Get rows of a table.

### Parameters:

(continued from last page)

`attributeXPath` - This path should point to one attribute cell.

**Returns:**

An array of table row xPaths.

---

## getTableRows

```
public java.util.ArrayList getTableRows(java.lang.String attributeXPath,  
    java.lang.String siblingXPath)
```

Get rows of a table.

**Parameters:**

`attributeXPath` - This path should point to one attribute cell.

`siblingXPath` - This path should point to the fact value cell of the attribute.

**Returns:**

An array of table row xPaths.

---

## getTableRows

```
public java.util.ArrayList getTableRows(org.w3c.dom.Document document,  
    java.lang.String attributeXPath,  
    java.lang.String siblingXPath)
```

Get rows of a table.

**Parameters:**

`document` - The document.

`attributeXPath` - This path should point to one attribute cell.

`siblingXPath` - This path should point to the fact value cell of the attribute.

**Returns:**

An array of table row xPaths.

---

## getNextTableRow

```
public java.lang.String getNextTableRow(java.lang.String XPath)
```

Find the next table row for a given XPath. For example: `//TABLE/TR[1]/TD[2] => //TABLE/TR[2]/TD[2]` `//TABLE/TR/TD[2] => //TABLE/TR[1]/TD[2]`

**Parameters:**

`xPath`

**Returns:**

## getParentNode

```
public static java.lang.String getParentNode(java.lang.String XPath)
```

Move one tag up in the DOM, e.g. `/div/span/a => /div/span`.

**Parameters:**

`xPath` - The XPath.

**Returns:**

The parent node.

---

---

## getNumberOfTableColumns

```
public int getNumberOfTableColumns(org.w3c.dom.Document document,  
    java.lang.String tableTDXPath)
```

Count the number of columns in a table.

### Parameters:

document - The document.  
tableTDXPath - The xPath to the table data tag.

### Returns:

The number of columns.

---

## getHTMLTextByXPath

```
public java.lang.String getHTMLTextByXPath(java.lang.String xPath)
```

---

## getTextByXPath

```
public java.lang.String getTextByXPath(java.lang.String xPath)
```

---

## getTextByXPath

```
public java.lang.String getTextByXPath(org.w3c.dom.Document document,  
    java.lang.String xpath)
```

---

## getTextsByXPath

```
public java.util.ArrayList getTextsByXPath(java.lang.String xPath)
```

If an xPath points to several (sibling) nodes, get the text of each node and add it to a list.

### Parameters:

xPath - The xPath.

### Returns:

A list of contents from the nodes that were targeted with the xPath.

---

## getTextsByXPath

```
public java.util.ArrayList getTextsByXPath(org.w3c.dom.Document document,  
    java.lang.String xpath)
```

---

## removeXPathIndices

```
public static java.lang.String removeXPathIndices(java.lang.String xPath)
```

---

---

## removeXPathIndices

```
public static java.lang.String removeXPathIndices(java.lang.String xpath,
    java.lang.String[] removeCountElements)
```

---

## removeXPathIndicesNot

```
public static java.lang.String removeXPathIndicesNot(java.lang.String xpath,
    java.lang.String[] notRemoveCountElements)
```

---

## printDOM

```
public static void printDOM(org.w3c.dom.Node node,
    java.lang.String indent)
```

---

## getTextDump

```
public static java.lang.String getTextDump(org.w3c.dom.Node node)
```

Get the sub tree as text.

**Parameters:**

node - The node from where to start.

**Returns:**

A string representation of the node and it's sub nodes.

---

## getHTMLText

```
public java.lang.String getHTMLText(org.w3c.dom.Node node)
```

---

## main

```
public static void main(java.lang.String[] args)
```

---



# tud.iir.extraction Class Query

```
java.lang.Object
└-- tud.iir.extraction.Query
```

Direct Known Subclasses:

[SnippetQuery](#)

---

```
public abstract class Query
extends java.lang.Object
```

Abstract Query class for entity, fact and snippet queries that are sent to a search engine.

**Author:**

David Urbansky

---

## Constructors

### Query

```
public Query()
```

## Methods

### getQueryType

```
public int getQueryType()
```

---

### setQueryType

```
public void setQueryType(int queryType)
```

---

### getQuerySet

```
public java.lang.String[] getQuerySet()
```

---

### setQuerySet

```
public void setQuerySet(java.lang.String[] querySet)
```

## tud.iir.extraction Class XPathSet

```
java.lang.Object
└--tud.iir.extraction.XPathSet
```

---

```
public class XPathSet
extends java.lang.Object
```

A set of xPaths.

**Author:**  
David Urbansky

---

### Constructors

#### XPathSet

```
public XPathSet()
```

### Methods

#### getXPathMap

```
public java.util.LinkedHashMap getXPathMap()
```

#### add

```
public void add(java.lang.String xPath)
```

#### addEntry

```
public void addEntry(java.util.Map.Entry entry)
```

#### getCountOfXPath

```
public int getCountOfXPath(java.lang.String xPath)
```

#### getHighestCountXPath

```
public java.lang.String getHighestCountXPath()
```

---

## getHighestCountXPath

```
public java.lang.String getHighestCountXPath(int minCount)
```

---

## getLongestHighCountXPath

```
public java.lang.String getLongestHighCountXPath(org.w3c.dom.Document document)
```

Return the longest (or highest priority) path that contains the highest count path as a substring.  
TODO b/a = a/b (website1.html)

**Returns:**

The longest xPath with the highest count.

---

Package

**tud.iir.extraction.content**

# tud.iir.extraction.content

## Class PageContentExtractor

```
java.lang.Object
```

```
└--tud.iir.extraction.content.PageContentExtractor
```

```
public class PageContentExtractor
extends java.lang.Object
```

A quick ~~and dirty~~ port of the JavaScript browser bookmarklet "Readability" by Arc90 -- a great tool for extracting content from HTML pages. *"Readability [...] takes a crack at wiping out all that junk so you can have a more enjoyable reading experience. [...] its success rate is pretty respectable (we'd guess over 90% of web sites are handled properly)".*

Note, that this is not designed for front pages like <http://cnn.com>, but for articles and blog entries with one topic. The result should be just the actual content, without irrelevant elements like navigation menus, headers, footers, ads, etc.

How it works, in a nutshell: Readability operates on the document's DOM tree. Basically, it assigns all elements a score for their contents. Metrics for the scoring are length of their text content, number of commas and link density. Also, "class" and "id" names are taken into consideration; for example, elements with class name "sidebar" contain unlikely actual content in contrast to elements with class "article". After the top element has been determined, the algorithm also checks its siblings whether they contain content, too.

### Author:

Philipp Katz, David Urbansky

### See Also:

[Website](#), [JavaScript Source](#)

### Version:

Based on: SVN r152, Jun 28, 2010

## Constructors

### PageContentExtractor

```
public PageContentExtractor()
```

## Methods

### setDocument

```
public PageContentExtractor setDocument(org.w3c.dom.Document document)
throws PageContentExtractorException
```

Set Document to be processed. Method returns *this* instance of PageContentExtractor, to allow convenient concatenations of method invocations, like: `new PageContentExtractor().setDocument(...).getResultDocument();`

#### Parameters:

document

#### Returns:

(continued from last page)

**Throws:**[PageContentExtractorException](#)

---

## setDocument

```
public PageContentExtractor setDocument(org.xml.sax.InputSource source)
    throws PageContentExtractorException
```

Set URL of document to be processed. Method returns `this` instance of `PageContentExtractor`, to allow convenient concatenations of method invocations, like: `new PageContentExtractor().setDocument(new URL(...)).getResultDocument();`

**Parameters:**`url`**Returns:****Throws:**[PageContentExtractorException](#)

---

## setDocument

```
public PageContentExtractor setDocument(java.net.URL url)
    throws PageContentExtractorException
```

Set URL of document to be processed. Method returns `this` instance of `PageContentExtractor`, to allow convenient concatenations of method invocations, like: `new PageContentExtractor().setDocument(new URL(...)).getResultDocument();`

**Parameters:**`url`**Returns:****Throws:**[PageContentExtractorException](#)

---

## setDocument

```
public PageContentExtractor setDocument(java.io.File file)
    throws PageContentExtractorException
```

Set File to be processed. Method returns `this` instance of `PageContentExtractor`, to allow convenient concatenations of method invocations, like: `new PageContentExtractor().setDocument(new File(...)).getResultDocument();`

**Parameters:**`file`**Returns:****Throws:**[PageContentExtractorException](#)

(continued from last page)

---

## setDocument

```
public PageContentExtractor setDocument(java.lang.String documentLocation)  
    throws PageContentExtractorException
```

Set the location of document to be processed. Method returns *this* instance of `PageContentExtractor`, to allow convenient concatenations of method invocations, like: `new PageContentExtractor().setDocument("http://website.com").getResultDocument();`

### Parameters:

`documentLocation` - The location of the document. This can be either a local file or a URL.

### Returns:

The instance of the `PageContentExtractor`.

### Throws:

[PageContentExtractorException](#)

---

## getResultDocument

```
public org.w3c.dom.Document getResultDocument()
```

Returns the filtered result document, as minimal XHTML fragment. Result just contains the filtered content, the result is not meant to be a complete web page or even to validate.

### Returns:

---

## getResultText

```
public java.lang.String getResultText()
```

Returns the filtered result as human readable plain text representation.

### Returns:

The extracted text from the document.

---

## getResultText

```
public java.lang.String getResultText(java.lang.String documentLocation)
```

Shortcut method for `new PageContentExtractor().setDocument("http://website.com").getResultText();`

### Parameters:

`documentLocation` - The location of the document. This can be either a local file or a URL.

### Returns:

The extracted text from the document.

---

## getResultTitle

```
public java.lang.String getResultTitle()
```

Returns the document's title. This will not just return the text from the document's `title` element, but try to remove generic, irrelevant substrings. For example, for a document with title *"Messi reveals close ties with Maradona - CNN.com"* this method will return *"Messi reveals close ties with Maradona"*.

### Returns:

(continued from last page)

---

## setWriteDump

```
public void setWriteDump(boolean writeDump)
```

Enable to write dumps of the DOM document with calculated weight.

### Parameters:

writeDump

---

## isWriteDump

```
public boolean isWriteDump()
```

---

## main

```
public static void main(java.lang.String[] args)  
    throws java.lang.Exception
```



# tud.iir.extraction.content

## Class PageContentExtractorException

```
java.lang.Object
├-- java.lang.Throwable
│   └-- java.lang.Exception
│       └-- tud.iir.extraction.content.PageContentExtractorException
```

All Implemented Interfaces:

java.io.Serializable

```
public class PageContentExtractorException
extends java.lang.Exception
```

## Constructors

### PageContentExtractorException

```
public PageContentExtractorException()
```

### PageContentExtractorException

```
public PageContentExtractorException(java.lang.Throwable t)
```

### PageContentExtractorException

```
public PageContentExtractorException(java.lang.String message,
                                     java.lang.Throwable cause)
```

### PageContentExtractorException

```
public PageContentExtractorException(java.lang.String message)
```

# tud.iir.extraction.content Class PreflightFilter

```
java.lang.Object
  |
  +--DefaultFilter
      |
      +--tud.iir.extraction.content.PreflightFilter
```

```
public class PreflightFilter
extends DefaultFilter
```

Filter out elements and attributes from the Document parsed with NekoHTML which can cause trouble later. This includes elements from foreign namespaces or attribute names with illegal characters.

**Author:**

Philipp Katz

## Constructors

### PreflightFilter

```
public PreflightFilter(Logger logger)
```

## Methods

### startElement

```
public void startElement(QName element,
    XMLAttributes attributes,
    Augmentations augs)
    throws XNIException
```

### emptyElement

```
public void emptyElement(QName element,
    XMLAttributes attributes,
    Augmentations augs)
    throws XNIException
```

### endElement

```
public void endElement(QName element,
    Augmentations augs)
    throws XNIException
```

---

Package

**tud.iir.extraction.entity**

# tud.iir.extraction.entity

## Class EntityDateComparator

java.lang.Object

└--tud.iir.extraction.entity.EntityDateComparator

### All Implemented Interfaces:

java.io.Serializable, java.util.Comparator

---

public class **EntityDateComparator**  
extends java.lang.Object  
implements java.util.Comparator, java.io.Serializable

Sort entities by the date they were last searched.

### Author:

David Urbansky

---

## Constructors

### EntityDateComparator

```
public EntityDateComparator()
```

## Methods

### compare

```
public int compare(Entity e1,  
                  Entity e2)
```

Sort that the oldest entity appears first.

#### Parameters:

e1 - Entity1

e2 - Entity2

# tud.iir.extraction.entity

## Class EntityExtractionProcess

```
java.lang.Object
  |
  +- java.lang.Thread
        |
        +- tud.iir.extraction.entity.EntityExtractionProcess
```

All Implemented Interfaces:  
java.lang.Runnable

---

```
public class EntityExtractionProcess
extends java.lang.Thread
```

The EntityExtractionProcess is a thread that runs the entity extraction.

**Author:**  
David Urbansky

---

## Constructors

### EntityExtractionProcess

```
public EntityExtractionProcess()
```

## Methods

### run

```
public void run()
```

---

### stopExtraction

```
public boolean stopExtraction()
```

# tud.iir.extraction.entity

## Class EntityExtractionThread

```
java.lang.Object
  |
  +- java.lang.Thread
        |
        +- tud.iir.extraction.entity.EntityExtractionThread
```

All Implemented Interfaces:  
java.lang.Runnable

```
public class EntityExtractionThread
extends java.lang.Thread
```

## Constructors

### EntityExtractionThread

```
public EntityExtractionThread(java.lang.ThreadGroup threadGroup,
                               java.lang.String name,
                               EntityExtractionTechnique entityExtractionTechnique,
                               EntityQuery entityQuery,
                               Concept concept,
                               java.lang.String url)
```

## Methods

### run

```
public void run()
```

# tud.iir.extraction.entity Class EntityExtractor

```
java.lang.Object
├── tud.iir.extraction.Extractor
│   └── tud.iir.extraction.entity.EntityExtractor
```

---

```
public class EntityExtractor
extends Extractor
```

The main class for the entity extraction. Here all three entity extraction techniques are triggered and called with the concept names.

**Author:**

David Urbansky

---

## Methods

### getInstance

```
public static Extractor getInstance()
```

---

### startExtraction

```
public void startExtraction(boolean phrase,
    boolean focusedCrawl,
    boolean seeds)
```

---

### startExtraction

```
public void startExtraction(boolean phrase,
    boolean focusedCrawl,
    boolean seeds,
    boolean continueFromLastExtraction)
```

---

### extractionFromPhrase

```
public void extractionFromPhrase()
```

Use simple generic patterns to extract entities from unstructured text.

---

### extractionFocusedCrawl

```
public void extractionFocusedCrawl()
```

Focused crawl extraction.

---

(continued from last page)

## extractionSeeds

```
public void extractionSeeds()
```

Extraction with seeds.

---

## extract

```
public void extract(EntityExtractionTechnique entityExtractionTechnique)
```

All entity extraction techniques use this method which handles threads, persistence management, querying, iterating through concepts and synonyms.

### Parameters:

`entityExtractionTechnique` - The entity extraction technique that should be used for a retrieved URL.

---

## getExtractions

```
public java.util.ArrayList getExtractions()
```

---

## printExtractions

```
public void printExtractions()
```

---

## createBenchmarkIndex

```
public void createBenchmarkIndex()
```

---

## isAutoSave

```
public boolean isAutoSave()
```

---

## setAutoSave

```
public void setAutoSave(boolean autoSave)
```

---

## getLogger

```
public Logger getLogger()
```

---

## getConcepts

```
public java.util.ArrayList getConcepts()
```

---



---

## setConcepts

```
public void setConcepts(java.util.ArrayList concepts)
```

---

## normalizeAllEntities

```
public void normalizeAllEntities()
```

---

## addExtraction

```
public void addExtraction(Entity newEntity)
```

---

## getExtractionLimit

```
public int getExtractionLimit()
```

---

## setExtractionLimit

```
public void setExtractionLimit(int extractionLimit)
```

---

## main

```
public static void main(java.lang.String[] a)
```

---

# tud.iir.extraction.entity

## Class EntityQueryFactory

```
java.lang.Object
└--tud.iir.extraction.entity.EntityQueryFactory
```

```
public class EntityQueryFactory
extends java.lang.Object
```

The EntityQueryFactory creates EntityQuery objects.

**Author:**  
David Urbansky

## Fields

### RETRIEVAL\_EXTRACTION\_TYPE\_PHRASE

```
public static final int RETRIEVAL_EXTRACTION_TYPE_PHRASE
```

Constant value: 1

### RETRIEVAL\_EXTRACTION\_TYPE\_FOCUSED\_CRAWL

```
public static final int RETRIEVAL_EXTRACTION_TYPE_FOCUSED_CRAWL
```

Constant value: 2

### RETRIEVAL\_EXTRACTION\_TYPE\_SEED

```
public static final int RETRIEVAL_EXTRACTION_TYPE_SEED
```

Constant value: 3

### TYPE\_XP\_SUCH\_AS

```
public static final int TYPE_XP_SUCH_AS
```

Constant value: 1

### TYPE\_XP\_LIKE

```
public static final int TYPE_XP_LIKE
```

Constant value: 2

### TYPE\_XP\_INCLUDING

```
public static final int TYPE_XP_INCLUDING
```

(continued from last page)

Constant value: 3

---

## TYPE\_XP\_ESPECIALLY

```
public static final int TYPE_XP_ESPECIALLY
```

Constant value: 4

---

## TYPE\_LIST\_OF\_XP

```
public static final int TYPE_LIST_OF_XP
```

Constant value: 5

---

## TYPE\_XS\_LIST

```
public static final int TYPE_XS_LIST
```

Constant value: 6

---

## TYPE\_BROWSE\_XP

```
public static final int TYPE_BROWSE_XP
```

Constant value: 8

---

## TYPE\_INDEX\_OF\_XP

```
public static final int TYPE_INDEX_OF_XP
```

Constant value: 9

---

## TYPE\_XS\_INDEX

```
public static final int TYPE_XS_INDEX
```

Constant value: 10

---

## TYPE\_SEED\_2

```
public static final int TYPE_SEED_2
```

Constant value: 11

---

## TYPE\_SEED\_3

```
public static final int TYPE_SEED_3
```

Constant value: 12

(continued from last page)

## TYPE\_SEED\_4

```
public static final int TYPE_SEED_4
```

Constant value: 13

## TYPE\_SEED\_5

```
public static final int TYPE_SEED_5
```

Constant value: 14

## Methods

### getInstance

```
public static EntityQueryFactory getInstance()
```

### getExtractionTypes

```
public static java.util.ArrayList getExtractionTypes()
```

### createPhraseQuery

```
public EntityQuery createPhraseQuery(Concept concept,  
                                       int type)
```

### createFocusedCrawlQuery

```
public EntityQuery createFocusedCrawlQuery(Concept concept,  
                                             int type)
```

### createSeedQuery

```
public EntityQuery createSeedQuery(Concept concept,  
                                     int type,  
                                     int numberOfCombinations)
```

# tud.iir.extraction.entity Class EntityTrustComparator

java.lang.Object

└--tud.iir.extraction.entity.EntityTrustComparator

## All Implemented Interfaces:

java.io.Serializable, java.util.Comparator

---

```
public class EntityTrustComparator
extends java.lang.Object
implements java.util.Comparator, java.io.Serializable
```

Sort entities by trust.

### Author:

David Urbansky

---

## Constructors

### EntityTrustComparator

```
public EntityTrustComparator()
```

## Methods

### compare

```
public int compare(Entity e1,
                  Entity e2)
```

Highest trust first.

#### Parameters:

e1 - Entity1

e2 - Entity2

# tud.iir.extraction.entity

## Class ListDiscoverer

```
java.lang.Object
|
|--tud.iir.extraction.entity.ListDiscoverer
```

```
public class ListDiscoverer
extends java.lang.Object
```

The ListDiscoverer tries to find a list (with entities) on a web page. If a "good" list is found the xPath for one or all entries in the list is returned. Features of a "good" list are as follows.

- the list has at least 10 entries
- it is the only long list on the web page, not just one of many (path lengths distribution)
- it has uniform entries, that is, entries are in almost the same format
- the list is specific for the web page, it should not be a navigation list that can be found on another page of the website

If no good list is found, an empty string is returned.

**Author:**

David Urbansky

## Constructors

### ListDiscoverer

```
public ListDiscoverer()
```

## Methods

### findPaginationURLs

```
public java.util.Set findPaginationURLs(org.w3c.dom.Document document)
```

### findPaginationURLs

```
public java.util.Set findPaginationURLs(java.lang.String url)
```

### findPaginationURLs

```
public java.util.Set findPaginationURLs()
```

### getPaginationURLs

```
public java.util.Set getPaginationURLs()
```

(continued from last page)

---

## getXPathSet

```
public XPathSet getXPathSet(org.w3c.dom.Document document)
```

Get a set of xPaths.

### Parameters:

`document` - The document the xPaths are constructed for.

### Returns:

A set of xPaths.

---

## discoverEntityXPath

```
public java.lang.String discoverEntityXPath(java.lang.String url)
```

---

## discoverEntityXPath

```
public java.lang.String discoverEntityXPath(org.w3c.dom.Document document)
```

---

## removeSiblingPagePaths

```
public XPathSet removeSiblingPagePaths(XPathSet xpathSet,  
    java.lang.String url,  
    org.w3c.dom.Document document)
```

---

## findEntityColumn

```
public int findEntityColumn(org.w3c.dom.Document document,  
    java.lang.String entityXPath)
```

---

## entriesUniform

```
public static boolean entriesUniform(java.util.ArrayList entries,  
    boolean tableDuplicateCheck)
```

Check whether a list of entries is likely to be a list of entities. The list is rejected if:

- more than 10% of them are just numbers
- more than 50% are only capitalized, e.g. CATEGORIES
- TODO does it make a difference?
- the average string length is more than 12 words
- there are not more than 10% entries that have duplicates
- there are not more than 10% entries missing

### Parameters:

`entries`

---

(continued from last page)

**Returns:**

True if the list entries are uniform, else false.

---

**getPaginationXPath**

```
public java.lang.String getPaginationXPath()
```

---

**setPaginationXPath**

```
public void setPaginationXPath(java.lang.String paginationXPath)
```

---

**main**

```
public static void main(java.lang.String[] args)
```

**Parameters:**

args



# tud.iir.extraction.entity

## Class PhraseExtractor

```
java.lang.Object
├-- tud.iir.extraction.entity.EntityExtractionTechnique
│   └-- tud.iir.extraction.entity.PhraseExtractor
```

---

```
public class PhraseExtractor
    extends EntityExtractionTechnique
```

The PhraseExtraction technique.

**Author:**

David Urbansky

---

## Constructors

### PhraseExtractor

```
public PhraseExtractor()
```

## Methods

### getPatterns

```
public java.lang.Integer[] getPatterns()
```

### getEntityQuery

```
public EntityQuery getEntityQuery(Concept concept,
                                     int entityType)
```

---

### extract

```
public void extract(java.lang.String url,
                    EntityQuery eq,
                    Concept concept)
```

# tud.iir.extraction.entity Class WrapperInductor

java.lang.Object

└-- tud.iir.extraction.entity WrapperInductor

All Implemented Interfaces:

[WrapperInductorInterface](#)

---

public abstract class **WrapperInductor**  
extends java.lang.Object  
implements [WrapperInductorInterface](#)

The abstract WrapperInductor class.

**Author:**

David Urbansky

---

## Constructors

### WrapperInductor

public **WrapperInductor**()

## Methods

### getExtractions

public java.util.ArrayList **getExtractions**()

## tud.iir.extraction.entity Interface WrapperInductorInterface

All Known Implementing Classes:

[WrapperInductor](#)

---

public interface **WrapperInductorInterface**  
extends

The WrapperInductorInterface class.

**Author:**

David Urbansky

---

### Methods

#### extract

```
public void extract(java.lang.String url,  
    EntityQuery eq,  
    Concept currentConcept)
```

## tud.iir.extraction.entity Class XPathAffixWrapper

```
java.lang.Object
  |
  +-- tud.iir.extraction.entity.AffixWrapper
        |
        +-- tud.iir.extraction.entity.XPathAffixWrapper
```

---

```
public class XPathAffixWrapper
extends AffixWrapper
```

The XPathAffixWrapper class.

**Author:**

David Urbansky

---

## Constructors

### XPathAffixWrapper

```
public XPathAffixWrapper(java.lang.String prefix,
                          java.lang.String suffix,
                          java.lang.String xPath)
```

## Methods

### getXPath

```
public java.lang.String getXPath()
```

**Returns:**

the xPath

---

### setXPath

```
public void setXPath(java.lang.String path)
```

**Parameters:**

path - the xPath to set

---

---

Package

**tud.iir.extraction.entity.ner**

# tud.iir.extraction.entity.ner Class Annotation

```
java.lang.Object
└-- tud.iir.extraction.entity.ner.Annotation
```

Direct Known Subclasses:  
[EvaluationAnnotation](#)

```
public class Annotation
extends java.lang.Object
```

An annotation made by a [NamedEntityRecognizer](#) when tagging a text.

**Author:**  
David Urbansky

## Constructors

### Annotation

```
public Annotation(Annotation annotation)
```

### Annotation

```
public Annotation(int offset,
                  java.lang.String entityName,
                  java.lang.String tagName)
```

### Annotation

```
public Annotation(int offset,
                  java.lang.String entityName,
                  CategoryEntries tags)
```

## Methods

### matches

```
public boolean matches(Annotation annotation)
```

### overlaps

```
public boolean overlaps(Annotation annotation)
```

---

## sameTag

```
public boolean sameTag(Annotation annotation)
```

---

---

## getOffset

```
public int getOffset()
```

---

---

## setOffset

```
public void setOffset(int offset)
```

---

---

## getLength

```
public int getLength()
```

---

---

## setLength

```
public void setLength(int length)
```

---

---

## getEndIndex

```
public int getEndIndex()
```

---

---

## getEntity

```
public Entity getEntity()
```

---

---

## setEntity

```
public void setEntity(Entity entity)
```

---

---

## getTags

```
public CategoryEntries getTags()
```

---

---

## setTags

```
public void setTags(CategoryEntries tags)
```

---

(continued from last page)

---

## getMostLikelyTag

```
public CategoryEntry getMostLikelyTag()
```

---

## getMostLikelyTagName

```
public java.lang.String getMostLikelyTagName()
```

---

## toString

```
public java.lang.String toString()
```



# tud.iir.extraction.entity.ner

## Class Annotations

```
java.lang.Object
├-- java.util.AbstractCollection
│   ├── java.util.AbstractList
│   │   ├── java.util.ArrayList
│   │   └-- tud.iir.extraction.entity.ner.Annotations
```

### All Implemented Interfaces:

java.util.Collection, java.util.List, java.io.Serializable, java.lang.Cloneable,  
java.util.RandomAccess, java.util.List

```
public class Annotations
extends java.util.ArrayList
```

A list of [AnnotationS](#).

**Author:**  
David Urbansky

## Constructors

### Annotations

```
public Annotations()
```

## Methods

### save

```
public void save(java.lang.String outputFilePath)
```

Save the annotation list to a file.

#### Parameters:

outputFilePath - The path where the annotation list should be saved to.

### sort

```
public void sort()
```

The order of annotations is important. Annotations are sorted by their offsets in ascending order.

### transformToEvaluationAnnotations

```
public void transformToEvaluationAnnotations()
```

# tud.iir.extraction.entity.ner

## Class FileFormatParser

```
java.lang.Object
└--tud.iir.extraction.entity.ner.FileFormatParser
```

---

```
public class FileFormatParser
extends java.lang.Object
```

Transform file formats for NER learning.

**Author:**

David Urbansky

---

## Constructors

### FileFormatParser

```
public FileFormatParser()
```

## Methods

### getText

```
public static java.lang.String getText(java.lang.String inputFilePath,
    TaggingFormat format)
```

---

### columnToXML

```
public static void columnToXML(java.lang.String inputFilePath,
    java.lang.String outputFilePath,
    java.lang.String columnSeparator)
```

Transform column format to XML. word [tab] type => <type>word</type>

**Parameters:**

inputFilePath - The location of the input file.

outputFilePath - The location where the transformed file should be written to.

columnSeparator - The separator for the columns.

---

### columnToBracket

```
public static void columnToBracket(java.lang.String inputFilePath,
    java.lang.String outputFilePath,
    java.lang.String columnSeparator)
```

---

(continued from last page)

## columnToColumnBIO

```
public static void columnToColumnBIO(java.lang.String inputFilePath,  
    java.lang.String outputFilePath,  
    java.lang.String columnSeparator)
```

---

## columnBIOToColumn

```
public static void columnBIOToColumn(java.lang.String inputFilePath,  
    java.lang.String outputFilePath,  
    java.lang.String columnSeparator)
```

---

## xmlToColumn

```
public static void xmlToColumn(java.lang.String inputFilePath,  
    java.lang.String outputFilePath,  
    java.lang.String columnSeparator)
```

---

## slashToXML

```
public static void slashToXML(java.lang.String slashFilePath,  
    java.lang.String xmlFilePath)
```

---

## slashToColumn

```
public static void slashToColumn(java.lang.String slashFilePath,  
    java.lang.String columnFilePath,  
    java.lang.String columnSeparator)
```

---

## bracketToXML

```
public static void bracketToXML(java.lang.String inputFilePath,  
    java.lang.String outputFilePath)
```

---

## bracketToColumn

```
public static void bracketToColumn(java.lang.String inputFilePath,  
    java.lang.String outputFilePath,  
    java.lang.String columnSeparator)
```

---

## columnTrainingToTest

```
public static void columnTrainingToTest(java.lang.String inputFilePath,  
    java.lang.String outputFilePath,  
    java.lang.String columnSeparator)
```

---

## tsvToSsv

```
public static void tsvToSsv(java.lang.String inputFilePath,  
                             java.lang.String outputFilePath)
```

---

## getAnnotations

```
public static Annotations getAnnotations(java.lang.String taggedTextFilePath,  
                                           TaggingFormat format)
```

---

## getAnnotationsFromColumn

```
public static Annotations getAnnotationsFromColumn(java.lang.String  
taggedTextFilePath)
```

---

## getAnnotationsFromXMLText

```
public static Annotations getAnnotationsFromXMLText(java.lang.String taggedText)
```

Get XML annotations from a text. Nested annotations are discarded.

### Parameters:

`taggedText` - The XML tagged text. For example "The <PHONE>iphone 4</PHONE> is a phone."

### Returns:

A list of annotations that were found in the text.

---

## getAnnotationsFromXMLFile

```
public static Annotations getAnnotationsFromXMLFile(java.lang.String  
taggedTextFilePath)
```

---

## main

```
public static void main(java.lang.String[] args)
```

### Parameters:

`args`

---

# tud.iir.extraction.entity.ner

## Class NamedEntityRecognizer

java.lang.Object

└─ tud.iir.extraction.entity.ner.NamedEntityRecognizer

Direct Known Subclasses:

[AlchemyNER](#), [IllinoisLbjNER](#), [LingPipeNER](#), [OpenCalaisNER](#), [OpenNLPNER](#), [StanfordNER](#), [TUDNER](#)

public abstract class **NamedEntityRecognizer**  
extends java.lang.Object

The abstract Named Entity Recognizer (NER). Every NER should provide functionality for tagging an input text. Some might also be able to be trained on input data.

**Author:**

David Urbansky

## Constructors

### NamedEntityRecognizer

public **NamedEntityRecognizer**()

## Methods

### getAnnotations

public abstract [Annotations](#) **getAnnotations**(java.lang.String inputText,  
java.lang.String configModelFilePath)

### getAnnotations

public [Annotations](#) **getAnnotations**(java.io.File inputTextFile,  
java.lang.String configModelFilePath)

### train

public abstract boolean **train**(java.lang.String trainingFilePath,  
java.lang.String modelFilePath)

Train the named entity recognizer using the data from the training file and save it to the model file path.

The training file must be given in tab separated column format where the first column is the term and the second column is the concept.

(continued from last page)

**Parameters:**

trainingFilePath - The path where the training data can be found.

modelFilePath - The path where the trained model should be saved to.

**Returns:**

True, if the training succeeded, false otherwise.

---

## train

```
public boolean train(java.io.File trainingFile,  
                    java.io.File modelFile)
```

---

## tag

```
public java.lang.String tag(java.lang.String inputText,  
                             java.lang.String configModelFilePath)
```

Tag the input text using the given model or model configuration.

**Parameters:**

inputText - The text to be tagged.

configModelFilePath - The file to the model or the configuration depending on the NER. Every NER has it's own model or configuration file.

**Returns:**The tagged string in the specified [TaggingFormat](#).

---

## tag

```
public java.lang.String tag(java.io.File inputFile,  
                             java.io.File configModelFile)
```

---

## tag

```
public void tag(java.lang.String inputText,  
                java.lang.String outputPath,  
                java.lang.String configModelFilePath)
```

---

## tag

```
public void tag(java.io.File inputFile,  
                java.io.File outputFile,  
                java.io.File configModelFile)
```

---

## evaluate

```
public EvaluationResult evaluate(java.lang.String testingFilePath,  
                                 java.lang.String configModelFilePath,  
                                 TaggingFormat format)
```

---

## printEvaluationDetails

```
public static java.lang.StringBuilder printEvaluationDetails(EvaluationResult
evaluationResult)
```

---

## printEvaluationDetails

```
public static java.lang.StringBuilder printEvaluationDetails(EvaluationResult
evaluationResult,
    java.lang.String targetPath)
```

---

## setName

```
public void setName(java.lang.String name)
```

---

## getName

```
public java.lang.String getName()
```

---

## setTaggingFormat

```
public void setTaggingFormat(TaggingFormat taggingFormat)
```

---

## getTaggingFormat

```
public TaggingFormat getTaggingFormat()
```

# tud.iir.extraction.entity.ner

## Class TaggingFormat

```

java.lang.Object
  |
  +- java.lang.Enum
        +- tud.iir.extraction.entity.ner.TaggingFormat
  
```

### All Implemented Interfaces:

java.io.Serializable, java.lang.Comparable

```

public final class TaggingFormat
extends java.lang.Enum
  
```

Different formats for named entity tagging a text.

### Author:

David Urbansky

## Fields

### XML

```
public static final tud.iir.extraction.entity.ner.TaggingFormat XML
```

Tag text with xml. For example: The Nexus One is expensive. => The <PHONE>Nexus One</PHONE> is expensive.

### COLUMN

```
public static final tud.iir.extraction.entity.ner.TaggingFormat COLUMN
```

Tag text in two columns where the first column is the token and the second is the tag. For example: The Nexus One is expensive. => The O Nexus PHONE One PHONE is O expensive O . O

### BRACKETS

```
public static final tud.iir.extraction.entity.ner.TaggingFormat BRACKETS
```

Tag text with brackets. For example: The Nexus One is expensive. => The [PHONE Nexus One ] is expensive.

### SLASHES

```
public static final tud.iir.extraction.entity.ner.TaggingFormat SLASHES
```

Tag text with brackets. For example: The Nexus One is expensive. => The Nexus/PHONE One/PHONE is expensive.

## Methods

### values

```
public static TaggingFormat\[\] values()
```



(continued from last page)

---

## valueOf

```
public static TaggingFormat valueOf(java.lang.String name)
```

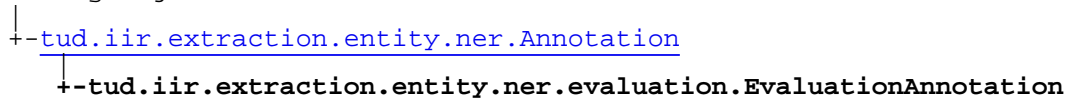
---

Package

**tud.iir.extraction.entity.ner.evaluation**

# tud.iir.extraction.entity.ner.evaluation Class EvaluationAnnotation

java.lang.Object



---

```
public class EvaluationAnnotation  
extends Annotation
```

---

## Constructors

### EvaluationAnnotation

```
public EvaluationAnnotation(Annotation annotation)
```

---

## Methods

### setTagged

```
public void setTagged(boolean tagged)
```

---

### isTagged

```
public boolean isTagged()
```

# tud.iir.extraction.entity.ner.evaluation

## Class EvaluationResult

java.lang.Object

└--tud.iir.extraction.entity.ner.evaluation.EvaluationResult

```
public class EvaluationResult
extends java.lang.Object
```

In NER there are 5 possible errors that can influence evaluation:

1. ERROR 1: tagged something that should not have been tagged
2. ERROR 2: missed an entity
3. ERROR 3: correct boundaries but wrong tag
4. ERROR 4: correctly tagged an entity but either too much or too little (wrong boundaries)
5. ERROR 5: wrong boundaries and wrong tag

We can evaluate using two approaches:

1. Exact match (`EvaluationResult.EXACT_MATCH`), that is, only if boundary and tag are assigned correctly, the assignment is true positive. Error types are not taken into account, all errors are equally wrong.
2. MUC (`EvaluationResult.MUC`), takes error types into account. 1 point for correct tag (regardless of boundaries), 1 point for correct text (regardless of tag). Totally correct (correct boundaries and correct tag) = 2 points

Author:

David Urbansky

## Fields

### EXACT\_MATCH

```
public static final int EXACT_MATCH
```

Constant value: 0

### MUC

```
public static final int MUC
```

Constant value: 1

### SPECIAL\_MARKER

```
public static final java.lang.String SPECIAL_MARKER
```

Constant value: #

### ERROR1

```
public static final java.lang.String ERROR1
```

(continued from last page)

Constant value: **#error1#**

---

## ERROR2

```
public static final java.lang.String ERROR2
```

Constant value: **#error2#**

---

## ERROR3

```
public static final java.lang.String ERROR3
```

Constant value: **#error3#**

---

## ERROR4

```
public static final java.lang.String ERROR4
```

Constant value: **#error4#**

---

## ERROR5

```
public static final java.lang.String ERROR5
```

Constant value: **#error5#**

---

## CORRECT

```
public static final java.lang.String CORRECT
```

Constant value: **#correct#**

---

## POSSIBLE

```
public static final java.lang.String POSSIBLE
```

Constant value: **#possible#**

## Constructors

## EvaluationResult

```
public EvaluationResult(java.util.Map assignments)
```

## Methods

(continued from last page)

## getPrecisionFor

```
public double getPrecisionFor(java.lang.String tagName,  
    int type)
```

---

## getRecallFor

```
public double getRecallFor(java.lang.String tagName,  
    int type)
```

---

## getF1For

```
public double getF1For(java.lang.String tagName,  
    int type)
```

---

## getTagAveragedPrecision

```
public double getTagAveragedPrecision(int type)
```

---

## getTagAveragedRecall

```
public double getTagAveragedRecall(int type)
```

---

## getTagAveragedF1

```
public double getTagAveragedF1(int type)
```

---

## getPrecision

```
public double getPrecision(int type)
```

---

## getRecall

```
public double getRecall(int type)
```

---

## getF1

```
public double getF1(int type)
```

---

(continued from last page)

## getAssignments

```
public java.util.Map getAssignments()
```

---

## setAssignments

```
public void setAssignments(java.util.Map assignments)
```

---

## toString

```
public java.lang.String toString()
```

---

Package

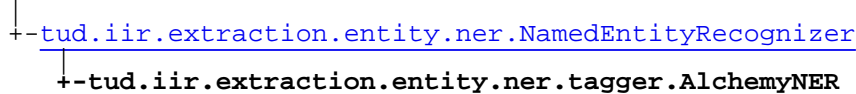
**tud.iir.extraction.entity.ner.tagger**



## tud.iir.extraction.entity.ner.tagger

### Class AlchemyNER

java.lang.Object



---

```
public class AlchemyNER
extends NamedEntityRecognizer
```

The Alchemy service for Named Entity Recognition. This class uses the Alchemy API and therefore requires the application to have access to the Internet.  
<http://www.alchemyapi.com/api/entity/textc.html>

Alchemy can recognize the following entities:

- Anniversary
- City
- Company
- Continent
- Country
- EntertainmentAward
- Facility
- FieldTerminology
- FinancialMarketIndex
- GeographicFeature
- HealthCondition
- Holiday
- Movie
- MusicGroup
- NaturalDisaster
- Organization
- Person
- PrintMedia
- RadioProgram
- RadioStation
- Region
- Sport
- StateOrCounty
- Technology
- TelevisionShow
- TelevisionStation
- AircraftManufacturer
- Airline
- AirportOperator
- ArchitectureFirm
- AutomobileCompany
- BicycleManufacturer
- BottledWater
- BreweryBrandOfBeer
- BroadcastDistributor
- CandyBarManufacturer
- ComicBookPublisher
- ComputerManufacturerBrand
- Distillery
- EngineeringFirm
- FashionLabel

- FilmCompany
- FilmDistributor
- GamePublisher
- ManufacturingPlant
- MusicalInstrumentCompany
- OperatingSystemDeveloper
- ProcessorManufacturer
- ProductionCompany
- RadioNetwork
- RecordLabel
- Restaurant
- RocketEngineDesigner
- RocketManufacturer
- ShipBuilder
- SoftwareDeveloper
- SpacecraftManufacturer
- SpiritBottler
- SpiritProductManufacturer
- TransportOperator
- TVNetwork
- VentureFundedCompany
- VentureInvestor
- VideoGameDeveloper
- VideoGameEngineDeveloper
- VideoGamePublisher
- WineProducer
- Airport
- Bridge
- HistoricPlace
- Hospital
- Lighthouse
- ShoppingMall
- SkiArea
- Skyscraper
- Stadium
- Station
- BodyOfWater
- Cave
- GeologicalFormation
- Glacier
- Island
- IslandGroup
- Lake
- Mountain
- MountainPass
- MountainRange
- OilField
- Park
- ProtectedArea
- River
- Waterfall
- Cave
- Island
- Lake
- Mountain
- Park
- ProtectedArea
- River
- TropicalCyclone
- AstronomicalSurveyProjectOrganization
- AwardPresentingOrganization
- Club
- CollegeUniversity

- CricketAdministrativeBody
- FinancialSupportProvider
- FootballOrganization
- FraternitySorority
- GovernmentAgency
- LegislativeCommittee
- Legislature
- MartialArtsOrganization
- MembershipOrganization
- NaturalOrCulturalPreservationAgency
- Non-ProfitOrganisation
- OrganizationCommittee
- PeriodicalPublisher
- PoliticalParty
- ReligiousOrder
- ReligiousOrganization
- ReportIssuingInstitution
- SoccerClub
- SpaceAgency
- SportsAssociation
- StudentOrganization
- TopLevelDomainRegistry
- TradeUnion
- FootballTeam
- HockeyTeam
- Legislature
- MilitaryUnit
- Non-ProfitOrganisation
- RecordLabel
- School
- SoccerClub
- TradeUnion
- Academic
- AircraftDesigner
- Appointee
- Architect
- ArchitectureFirmPartner
- Astronaut
- Astronomer
- Author
- AutomotiveDesigner
- AwardJudge
- AwardNominee
- AwardWinner
- BasketballCoach
- BasketballPlayer
- Bassist
- Blogger
- BoardMember
- Boxer
- BroadcastArtist
- Celebrity
- Chef
- ChessPlayer
- ChivalricOrderFounder
- ChivalricOrderMember
- ChivalricOrderOfficer
- Collector
- ComicBookColorist
- ComicBookCreator
- ComicBookEditor
- ComicBookInker
- ComicBookLetterer

- ComicBookPenciler
- ComicBookWriter
- ComicStripArtist
- ComicStripCharacter
- ComicStripCreator
- CompanyAdvisor
- CompanyFounder
- CompanyShareholder
- Composer
- ComputerDesigner
- ComputerScientist
- ConductedEnsemble
- Conductor
- CricketBowler
- CricketCoach
- CricketPlayer
- CricketUmpire
- Cyclist
- Dedicatee
- Dedicator
- Deity
- DietFollower
- DisasterSurvivor
- DisasterVictim
- Drummer
- ElementDiscoverer
- FashionDesigner
- FictionalCreature
- FictionalUniverseCreator
- FilmActor
- FilmArtDirector
- FilmCastingDirector
- FilmCharacter
- FilmCinematographer
- FilmCostumerDesigner
- FilmCrewmember
- FilmCritic
- FilmDirector
- FilmEditor
- FilmMusicContributor
- FilmProducer
- FilmProductionDesigner
- FilmSetDesigner
- FilmTheorist
- FilmWriter
- FootballCoach
- FootballPlayer
- FootballReferee
- FootballTeamManager
- FoundingFigure
- GameDesigner
- Golfer
- Guitarist
- HallOfFameInductee
- Hobbyist
- HockeyCoach
- HockeyPlayer
- HonoraryDegreeRecipient
- Illustrator
- Interviewer
- Inventor
- LandscapeArchitect
- LanguageCreator

- Lyricist
- MartialArtist
- MilitaryCommander
- MilitaryPerson
- Monarch
- Mountaineer
- MusicalArtist
- MusicalGroupMember
- NoblePerson
- NobleTitle
- OlympicAthlete
- OperaCharacter
- OperaDirector
- OperaLibretto
- OperaSinger
- PeriodicalEditor
- Physician
- PoliticalAppointer
- Politician
- ProAthlete
- ProgrammingLanguageDesigner
- ProgrammingLanguageDeveloper
- ProjectParticipant
- RecordingEngineer
- RecordProducer
- ReligiousLeader
- SchoolFounder
- ShipDesigner
- Songwriter
- SportsLeagueAwardWinner
- SportsOfficial
- Surgeon
- TennisPlayer
- TennisTournamentChampion
- TheaterActor
- TheaterCharacter
- TheaterChoreographer
- TheaterDesigner
- TheaterDirector
- TheaterProducer
- TheatricalComposer
- TheatricalLyricist
- Translator
- TVActor
- TVCharacter
- TVDirector
- TVPersonality
- TVProducer
- TVProgramCreator
- TVWriter
- U.S.Congressperson
- USPresident
- USVicePresident
- VideoGameActor
- VideoGameDesigner
- VisualArtist
- Actor
- Architect
- Astronaut
- Athlete
- BritishRoyalty
- Cardinal
- ChristianBishop

- CollegeCoach
- Comedian
- ComicsCreator
- Congressman
- Criminal
- FootballManager
- Journalist
- MilitaryPerson
- Model
- Monarch
- MusicalArtist
- Philosopher
- Politician
- Saint
- Scientist
- Writer
- Magazine
- Newspaper
- SchoolNewspaper
- EnglishRegion
- FrenchRegion
- ItalianRegion
- VideoGameRegion
- WineRegion
- MartialArt
- PoliticalDistrict
- AdministrativeDivision
- GovernmentalJurisdiction

See also <http://www.alchemyapi.com/api/entity/types.html>

Author:

David Urbansky

---

## Constructors

### AlchemyNER

```
public AlchemyNER()
```

## Methods

### train

```
public boolean train(java.lang.String trainingFilePath,  
                    java.lang.String modelFilePath)
```

Train the named entity recognizer using the data from the training file and save it to the model file path.

The training file must be given in tab separated column format where the first column is the term and the second column is the concept.

---

### getAnnotations

```
public Annotations getAnnotations(java.lang.String inputText,  
                                java.lang.String configModelFilePath)
```

(continued from last page)

---

## tag

```
public java.lang.String tag(java.lang.String inputText)
```

Tag the input text. Alchemy API does not require to specify a model.

**Parameters:**

`inputText` - The text to be tagged.

**Returns:**

The tagged text.

---

## main

```
public static void main(java.lang.String[] args)
```

# tud.iir.extraction.entity.ner.tagger

## Class IllinoisLbjNER

```
java.lang.Object
├── tud.iir.extraction.entity.ner.NamedEntityRecognizer
│   └── tud.iir.extraction.entity.ner.tagger.IllinoisLbjNER
```

---

```
public class IllinoisLbjNER
extends NamedEntityRecognizer
```

This class wraps the Learning Java Based Illinois Named Entity Tagger. The implementation is in an external library and the approach is explained in the following paper by L. Ratinov and D. Roth: "Design Challenges and Misconceptions in Named Entity Recognition", CoNLL 2009

See also <http://l2r.cs.uiuc.edu/~cogcomp/asoftware.php?skey=FLBJNE>

Author:  
David Urbansky

---

## Constructors

### IllinoisLbjNER

```
public IllinoisLbjNER()
```

## Methods

### demo

```
public void demo(boolean forceSentenceSplitsOnNewLines,
                 java.lang.String configFilePath)
    throws java.io.IOException
```

---

### train

```
public boolean train(java.lang.String trainingFilePath,
                    java.lang.String modelFilePath)
```

Train the named entity recognizer using the data from the training file and save it to the model file path.

The training file must be given in tab separated column format where the first column is the term and the second column is the concept.

---

### getAnnotations

```
public Annotations getAnnotations(java.lang.String inputText,
                                 java.lang.String configModelFilePath)
```

---



(continued from last page)

---

## trainNER

```
public void trainNER(java.lang.String trainingFilePath,  
    java.lang.String testingFilePath,  
    boolean forceSentenceSplitsOnNewLines,  
    java.lang.String configFilePath)
```

---

## testNER

```
public void testNER(java.lang.String testingFilePath,  
    boolean forceSentenceSplitsOnNewLines,  
    java.lang.String configFilePath)
```

---

## useLearnedNER

```
public void useLearnedNER(java.lang.String inputText,  
    boolean forceSentenceSplitsOnNewLines,  
    java.lang.String configFilePath)
```

---

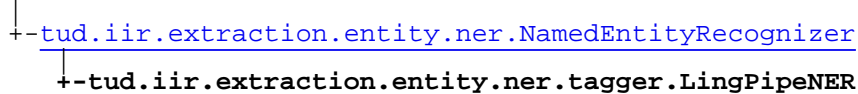
## main

```
public static void main(java.lang.String[] args)
```

# tud.iir.extraction.entity.ner.tagger

## Class LingPipeNER

java.lang.Object



---

```
public class LingPipeNER
extends NamedEntityRecognizer
```

This class wraps the LingPipe implementation of a Named Entity Recognizer.

See also <http://alias-i.com/lingpipe/demos/tutorial/ne/read-me.html>

Author:

David Urbansky

---

## Constructors

### LingPipeNER

```
public LingPipeNER()
```

## Methods

### train

```
public boolean train(java.lang.String trainingFilePath,
    java.lang.String modelFilePath)
```

Train the named entity recognizer using the data from the training file and save it to the model file path.

The training file must be given in tab separated column format where the first column is the term and the second column is the concept.

---

### getAnnotations

```
public Annotations getAnnotations(java.lang.String inputText,
    java.lang.String configModelFilePath)
```

---

### trainNER

```
public void trainNER(java.lang.String trainingFilePath,
    java.lang.String developmentFilePath,
    java.lang.String modelOutputFilePath)
throws java.io.IOException
```

---

## evaluateNER

```
public void evaluateNER(java.lang.String modelFilePath,  
    java.lang.String testFilePath)  
    throws java.lang.Exception
```

---

## scoreNER

```
public void scoreNER(java.lang.String[] args)  
    throws java.io.IOException
```

---

## useLearnedNER

```
public void useLearnedNER(java.lang.String modelFilePath,  
    java.lang.String inputText)  
    throws java.io.IOException,  
        java.lang.ClassNotFoundException
```

---

## main

```
public static void main(java.lang.String[] args)
```

## tud.iir.extraction.entity.ner.tagger Class OpenCalaisNER

```
java.lang.Object
├── tud.iir.extraction.entity.ner.NamedEntityRecognizer
│   └── tud.iir.extraction.entity.ner.tagger.OpenCalaisNER
```

---

```
public class OpenCalaisNER
extends NamedEntityRecognizer
```

The Open Calais service for Named Entity Recognition. This class uses the Open Calais API and therefore requires the application to have access to the Internet.

Open Calais can recognize the following entities:

- Anniversary
- City
- Company
- Continent
- Country
- Currency
- EmailAddress
- EntertainmentAwardEvent
- Facility
- FaxNumber
- Holiday
- IndustryTerm
- MarketIndex
- MedicalCondition
- MedicalTreatment
- Movie
- MusicAlbum
- MusicGroup
- NaturalFeature
- OperatingSystem
- Organization
- Person
- PhoneNumber
- PoliticalEvent
- Position
- Product
- ProgrammingLanguage
- ProvinceOrState
- PublishedMedium
- RadioProgram
- RadioStation
- Region
- SportsEvent
- SportsGame
- SportsLeague
- Technology
- TVShow
- TVStation
- URL

See also <http://www.opencalais.com/documentation/calais-web-service-api/api-metadata/entity-index-and-definitions>

**Author:**David Urbansky

---

## Constructors

### OpenCalaisNER

```
public OpenCalaisNER()
```

## Methods

### train

```
public boolean train(java.lang.String trainingFilePath,  
                    java.lang.String modelFilePath)
```

Train the named entity recognizer using the data from the training file and save it to the model file path.

The training file must be given in tab separated column format where the first column is the term and the second column is the concept.

---

### getAnnotations

```
public Annotations getAnnotations(java.lang.String inputText,  
                                  java.lang.String configModelFilePath)
```

---

### tag

```
public java.lang.String tag(java.lang.String inputText)
```

Tag the input text. Open Calais does not require to specify a model.

**Parameters:**

`inputText` - The text to be tagged.

**Returns:**

The tagged text.

---

### main

```
public static void main(java.lang.String[] args)
```

# tud.iir.extraction.entity.ner.tagger

## Class OpenNLPNER

```
java.lang.Object
├── tud.iir.extraction.entity.ner.NamedEntityRecognizer
│   └── tud.iir.extraction.entity.ner.tagger.OpenNLPNER
```

---

```
public class OpenNLPNER
extends NamedEntityRecognizer
```

This class wraps the OpenNLP Named Entity Recognizer which uses a maximum entropy approach.

The following models exist already for this recognizer:

- Date
- Location
- Money
- Organization
- Percentage
- Person
- Time

Changes to the original OpenNLP code:

- made nameFinder public in NameFinder.java
- NameSampleDataStream.java added lines 43 to 46 to allow non white-spaced tagging
- the model names must have the following format openNLP\_TAG.bin.gz where "TAG" is the name of the tag that will be tagged by this model

See also

[http://sourceforge.net/apps/mediawiki/opennlp/index.php?title=Name\\_Finder#Named\\_Entity\\_Annotation\\_Guidelines](http://sourceforge.net/apps/mediawiki/opennlp/index.php?title=Name_Finder#Named_Entity_Annotation_Guidelines)

**Author:**

David Urbansky

---

## Constructors

### OpenNLPNER

```
public OpenNLPNER()
```

## Methods

### demo

```
public void demo()
```

## demo

```
public void demo(java.lang.String inputText)
```

---

## getAnnotations

```
public Annotations getAnnotations(java.lang.String inputText,  
    java.lang.String configModelFilePath)
```

---

## train

```
public boolean train(java.lang.String trainingFilePath,  
    java.lang.String modelFilePath)
```

Train the named entity recognizer using the data from the training file and save it to the model file path.

The training file must be given in tab separated column format where the first column is the term and the second column is the concept.

---

## main

```
public static void main(java.lang.String[] args)  
    throws java.lang.Exception
```

### Parameters:

args

### Throws:

Exception

# tud.iir.extraction.entity.ner.tagger

## Class StanfordNER

```
java.lang.Object
├── tud.iir.extraction.entity.ner.NamedEntityRecognizer
│   └── tud.iir.extraction.entity.ner.tagger.StanfordNER
```

---

```
public class StanfordNER
    extends NamedEntityRecognizer
```

This class wraps the Stanford Named Entity Recognizer which is based on conditional random fields (CRF).

The NER has been described in the following paper:

The following models exist already for this recognizer:

- Person
- Location
- Organization

Jenny Rose Finkel, Trond Grenager, and Christopher Manning  
"Incorporating Non-local Information into Information Extraction Systems", 2005  
Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005),  
pp. 363-370  
[Read Paper](#)

See also <http://www-nlp.stanford.edu/software/crf-faq.shtml>

**Author:**

David Urbansky

---

## Constructors

### StanfordNER

```
public StanfordNER()
```

## Methods

### demo

```
public void demo(java.lang.String inputText)
    throws java.io.IOException
```

---

### train

```
public boolean train(java.lang.String trainingFilePath,
    java.lang.String modelFilePath)
```



(continued from last page)

Train the named entity recognizer using the data from the training file and save it to the model file path.

The training file must be given in tab separated column format where the first column is the term and the second column is the concept.

---

## getAnnotations

```
public Annotations getAnnotations(java.lang.String inputText,  
    java.lang.String configModelFilePath)
```

---

## useLearnedNER

```
public void useLearnedNER(java.lang.String modelFilePath,  
    java.lang.String inputText)  
    throws java.io.IOException
```

---

## trainNER

```
public void trainNER(java.lang.String configFilePath)  
    throws java.lang.Exception
```

---

## evaluateNER

```
public void evaluateNER(java.lang.String modelFilePath,  
    java.lang.String testFilePath)  
    throws java.lang.Exception
```

---

## main

```
public static void main(java.lang.String[] args)  
    throws java.lang.Exception
```

### Parameters:

args

### Throws:

Exception

# tud.iir.extraction.entity.ner.tagger

## Class TUDNER

java.lang.Object

└-[tud.iir.extraction.entity.ner.NamedEntityRecognizer](#)  
└-tud.iir.extraction.entity.ner.tagger.TUDNER

All Implemented Interfaces:  
java.io.Serializable

---

public class **TUDNER**  
extends [NamedEntityRecognizer](#)  
implements java.io.Serializable

---

## Constructors

### TUDNER

public **TUDNER**()

## Methods

### train

public boolean **train**(java.lang.String trainingFilePath,  
java.lang.String modelFilePath)

Train the named entity recognizer using the data from the training file and save it to the model file path.

The training file must be given in tab separated column format where the first column is the term and the second column is the concept.

---

### getAnnotations

public [Annotations](#) **getAnnotations**(java.lang.String inputText,  
java.lang.String modelPath)

---

### getTrainingEntities

public [EntityList](#) **getTrainingEntities**(double percentage)

---

(continued from last page)

## load

```
public void load(java.lang.String modelPath)
```

---

## getKbCommunicator

```
public KnowledgeBaseCommunicatorInterface getKbCommunicator()
```

---

## setKbCommunicator

```
public void setKbCommunicator(KnowledgeBaseCommunicatorInterface kbCommunicator)
```

---

## main

```
public static void main(java.lang.String[] args)
```

### Parameters:

args

---

Package

**tud.iir.extraction.event**

# tud.iir.extraction.event

## Class Event

```
java.lang.Object
├── tud.iir.knowledge.Extractable
│   └── tud.iir.extraction.event.Event
```

All Implemented Interfaces:  
java.io.Serializable

public class **Event**  
extends [Extractable](#)

Author:  
Martin Wunderwald

## Constructors

### Event

```
public Event()
```

### Event

```
public Event(java.lang.String url)
```

### Event

```
public Event(java.lang.String title,  
             java.lang.String text)
```

### Event

```
public Event(java.lang.String title,  
             java.lang.String text,  
             java.lang.String url)
```

## Methods

### getTitle

```
public java.lang.String getTitle()
```

---

## setTitle

```
public void setTitle(java.lang.String title)
```

---

---

## getFeatures

```
public FeatureObject getFeatures()
```

---

---

## setFeatures

```
public void setFeatures(FeatureObject features)
```

---

---

## getEntityFeatures

```
public java.util.Map getEntityFeatures()
```

---

---

## setEntityFeatures

```
public void setEntityFeatures(java.util.Map entityFeatures)
```

---

---

## getEntityChunks

```
public java.util.Map getEntityChunks()
```

---

---

## setEntityChunks

```
public void setEntityChunks(java.util.Map entityChunks)
```

---

---

## getUrl

```
public java.lang.String getUrl()
```

---

---

## setUrl

```
public void setUrl(java.lang.String url)
```

---

---

## getWebresults

```
public java.util.List getWebresults()
```

---

(continued from last page)

---

## setWebresults

```
public void setWebresults(java.util.List webresult)
```

---

---

## getText

```
public java.lang.String getText()
```

---

---

## setText

```
public void setText(java.lang.String text)
```

---

---

## getWho

```
public java.lang.String getWho()
```

---

---

## setWho

```
public void setWho(java.lang.String who)
```

---

---

## getWhere

```
public java.lang.String getWhere()
```

---

---

## setWhere

```
public void setWhere(java.lang.String where)
```

---

---

## getWhat

```
public java.lang.String getWhat()
```

---

---

## setWhat

```
public void setWhat(java.lang.String what)
```

---

(continued from last page)

---

## getWhy

```
public java.lang.String getWhy()
```

---

## setWhy

```
public void setWhy(java.lang.String why)
```

---

## getWhen

```
public java.lang.String getWhen()
```

---

## setWhen

```
public void setWhen(java.lang.String when)
```

---

## getHow

```
public java.lang.String getHow()
```

---

## setHow

```
public void setHow(java.lang.String how)
```



# tud.iir.extraction.event

## Class EventAggregator

```
java.lang.Object
└-- tud.iir.extraction.event.EventAggregator
```

---

```
public class EventAggregator
extends java.lang.Object
```

**Author:**  
Martin Wunderwald

---

## Constructors

### EventAggregator

```
public EventAggregator()
```

## Methods

### aggregate

```
public void aggregate()
```

---

### getEventmap

```
public java.util.Map getEventmap()
```

---

### setMaxThreads

```
public void setMaxThreads(int maxThreads)
```

Sets the maximum number of parallel threads when aggregating or adding multiple new feeds.

**Parameters:**

maxThreads

---

### getEvents

```
public java.util.List getEvents()
```

---

### setEvents

```
public void setEvents(java.util.List events)
```

---

(continued from last page)

---

## getQuery

```
public java.lang.String getQuery()
```

---

## setQuery

```
public void setQuery(java.lang.String query)
```

---

## getResultCount

```
public int getResultCount()
```

---

## setResultCount

```
public void setResultCount(int resultCount)
```

---

## getMaxThreads

```
public int getMaxThreads()
```

---

## main

```
public static void main(java.lang.String[] args)
```

## tud.iir.extraction.event Class EventAggregatorException

```
java.lang.Object
  |
+- java.lang.Throwable
  |
+- java.lang.Exception
    |
    +- tud.iir.extraction.event.EventAggregatorException
```

All Implemented Interfaces:  
java.io.Serializable

---

```
public class EventAggregatorException
extends java.lang.Exception
```

Author:  
Martin Wunderwald

---

### Constructors

#### EventAggregatorException

```
public EventAggregatorException(java.lang.Throwable throwable)
```

---

#### EventAggregatorException

```
public EventAggregatorException(java.lang.String string)
```

# tud.iir.extraction.event

## Class EventExtractor

```
java.lang.Object
├── tud.iir.extraction.Extractor
│   └── tud.iir.extraction.event.EventExtractor
```

---

```
public class EventExtractor
    extends Extractor
```

Event Extractor  
**Author:**  
Martin Wunderwald

---

## Methods

### getInstance

```
public static Extractor getInstance()
```

**Returns:**  
EventExtractor

---

### startExtraction

```
public void startExtraction()
```

---

### startExtraction

```
public void startExtraction(boolean continueFromLastExtraction)
```

---

### extractEventFromURL

```
public static Event extractEventFromURL(java.lang.String url)
```

extracts an event from given url

**Parameters:**  
url - - url of a news article

**Returns:**  
Event - The event

---

### main

```
public static void main(java.lang.String[] args)
```

---

(continued from last page)

**Parameters:**

args

# tud.iir.extraction.event

## Class EventFeatureExtractor

```
java.lang.Object
└--tud.iir.extraction.event.EventFeatureExtractor
```

```
public class EventFeatureExtractor
extends java.lang.Object
```

EventFeatureExtractor to extract Features from Events

**Author:**  
Martin Wunderwald

## Constructors

### EventFeatureExtractor

```
public EventFeatureExtractor()
```

## Methods

### setFeatures

```
public static void setFeatures(Event event)
```

sets the features of an event

**Parameters:**  
event

### aggregateEvents

```
public static java.util.Map aggregateEvents(java.lang.String query)
```

aggregates events from SearchEngines by a given query

**Parameters:**  
query - - the query

**Returns:**

### writeCSV

```
public static void writeCSV(java.util.Map eventMap,
    java.util.List whos,
    java.util.List wheres,
    java.util.List whats,
    boolean append)
```

writes events to CSV file for training the classifier

(continued from last page)

**Parameters:**

eventMap  
whos  
wheres  
whats  
append

---

**main**

```
public static void main(java.lang.String[] args)
```

**Parameters:**

args

# tud.iir.extraction.event

## Class PhraseChunker

```
java.lang.Object
└--tud.iir.extraction.event.PhraseChunker
```

```
public class PhraseChunker
extends java.lang.Object
```

Expects to chunk 1 sentence at a time.

**Author:**  
Martin Wunderwald

## Constructors

### PhraseChunker

```
public PhraseChunker(HmmDecoder postTagger,
                     TokenizerFactory tokenizerFactory)
```

## Methods

### chunk

```
public Chunking chunk(java.lang.CharSequence cSeq)
```

### chunk

```
public Chunking chunk(char[] cs,
                     int start,
                     int end)
```

### main

```
public static void main(java.lang.String[] args)
```



# tud.iir.extraction.event Class WhereClassifier

```
java.lang.Object
├── tud.iir.classification.Classifier
│   └── tud.iir.extraction.event.WhereClassifier
```

---

```
public class WhereClassifier
extends Classifier
```

**Author:**  
Martin Wunderwald

---

## Constructors

### WhereClassifier

```
public WhereClassifier(int type)
```

## Methods

### classify

```
public float classify(FeatureObject fo)
```

**Parameters:**  
fo

**Returns:**

---

### useTrainedClassifier

```
public void useTrainedClassifier()
```

Use an already trained classifier.

---

### main

```
public static void main(java.lang.String[] args)
```

**Parameters:**  
args

---

Package

**tud.iir.extraction.fact**

# tud.iir.extraction.fact

## Class EntityFactExtractionThread

```
java.lang.Object
  |
  +- java.lang.Thread
        |
        +- tud.iir.extraction.fact.EntityFactExtractionThread
```

All Implemented Interfaces:  
java.lang.Runnable

```
public class EntityFactExtractionThread
extends java.lang.Thread
```

The EntityFactExtractionThread extracts facts for one given entity. Therefore, extracting facts can be parallelized on the entity level.

**Author:**  
David Urbansky

## Constructors

### EntityFactExtractionThread

```
public EntityFactExtractionThread(java.lang.ThreadGroup threadGroup,
                                   java.lang.String name,
                                   Entity entity)
```

## Methods

### run

```
public void run()
```

### getSource

```
public java.lang.String getSource()
```

### setSource

```
public void setSource(java.lang.String source)
```

# tud.iir.extraction.fact Class FactExtractionDecisionTree

java.lang.Object

└- tud.iir.extraction.fact.FactExtractionDecisionTree

```
public class FactExtractionDecisionTree
extends java.lang.Object
```

The fact extraction decision tree creates a DOM of a given mark up and searches for a given attribute depending on where the attribute is, a decision about the corresponding value will be made. e.g. whether attribute is in a table or in free text

**Author:**

David Urbansky

## Constructors

### FactExtractionDecisionTree

```
public FactExtractionDecisionTree(Entity entity,
                                   java.lang.String url)
```

### FactExtractionDecisionTree

```
public FactExtractionDecisionTree(Entity entity,
                                   java.lang.String url,
                                   Attribute attribute)
```

## Methods

### setDocument

```
public void setDocument(java.lang.String url)
```

### getEntity

```
public Entity getEntity()
```

### setEntity

```
public void setEntity(Entity entity)
```

(continued from last page)

## getAttribute

```
public Attribute getAttribute()
```

---

## setAttribute

```
public void setAttribute(Attribute attribute)
```

---

## getFactStrings

```
public java.util.HashMap getFactStrings(Attribute attribute)
```

Run the decision tree and find the string where the fact value for the given attribute is most likely to be found extract the value and add it to the entity facts (fact values).

**Parameters:**

`attribute` - The initial attribute.

**Returns:**

All strings with the values for the given attribute.

---

## main

```
public static void main(java.lang.String[] args)
```

**Parameters:**

`args`

# tud.iir.extraction.fact

## Class FactExtractionProcess

```
java.lang.Object
├-- java.lang.Thread
│   └-- tud.iir.extraction.fact.FactExtractionProcess
```

All Implemented Interfaces:  
java.lang.Runnable

```
public class FactExtractionProcess
extends java.lang.Thread
```

The fact extraction process.

**Author:**  
David Urbansky

## Constructors

### FactExtractionProcess

```
public FactExtractionProcess()
```

### FactExtractionProcess

```
public FactExtractionProcess(boolean benchmark)
```

## Methods

### run

```
public void run()
```

### stopExtraction

```
public boolean stopExtraction()
```

### isBenchmark

```
public boolean isBenchmark()
```

(continued from last page)

## setBenchmark

```
public void setBenchmark(boolean benchmark)
```

# tud.iir.extraction.fact

## Class FactExtractor

```
java.lang.Object
├── tud.iir.extraction.Extractor
│   └── tud.iir.extraction.fact.FactExtractor
```

```
public class FactExtractor
extends Extractor
```

The FactExtractor class. This class is singleton.

**Author:**

David Urbansky

## Methods

### getInstance

```
public static Extractor getInstance()
```

### extractFactsForEntityName

```
public java.util.ArrayList extractFactsForEntityName(java.lang.String entityName)
```

This methods allows it to extract facts (attribute - value pairs) for a given entity name, for which the concept is unknown.

**Parameters:**

`entityName` - The name of the entity, facts are searched for.

**Returns:**

An array of extracted facts.

### startExtraction

```
public void startExtraction()
```

Start extraction of facts for entities that are fetched from the knowledge base. Continue from last extraction.

### startExtraction

```
public void startExtraction(boolean continueFromLastExtraction)
```

### createFactLog

```
public void createFactLog()
```

Log which facts for which concepts and entities have been extracted.



## createFactLog

```
public void createFactLog(java.lang.String headText)
```

---

## extractFacts

```
public static java.util.ArrayList extractFacts(java.lang.String url,  
        java.util.HashSet attributes)
```

Try to find facts from a table on a web page. Use a set of attribute names to detect the table and the other facts. If no attributes are given facts are tried to be extracted without any prior information.

**Parameters:**

url - The URL of the web page.

attributes - A set of attribute names that appear (on the page AND) in the table.

**Returns:**

A list of facts that were found in the table.

---

## extractFacts

```
public static java.util.ArrayList extractFacts(java.lang.String url)
```

---

## main

```
public static void main(java.lang.String[] args)
```

Example calls of fact extraction functionality. See `FactExtractionTest` for more tests and usages.

**Parameters:**

args

## tud.iir.extraction.fact Class FactString

```
java.lang.Object
|
|--tud.iir.extraction.fact.FactString
```

---

```
public class FactString
extends java.lang.Object
```

The fact string is the string where the fact value is expected to be found in this string can have been derived from different methods depending on where the attribute has been found e.g. if in free text "The i8510 INNOV8 offers 16GB of built-in memory in addition to a microSD card slot for even more storage options" or in a colon of a table "16GB internal, microSD card slot" the distinction is important as the value extraction can differ for these types

**Author:**

David Urbansky

---

### Constructors

#### FactString

```
public FactString(java.lang.String factString,
                  int extractionType)
```

### Methods

#### getFactString

```
public java.lang.String getFactString()
```

---

#### setFactString

```
public void setFactString(java.lang.String factString)
```

---

#### getExtractionType

```
public int getExtractionType()
```

---

#### setExtractionType

```
public void setExtractionType(int type)
```

---

(continued from last page)

## toString

```
public java.lang.String toString()
```

# tud.iir.extraction.fact

## Class FactValueComparator

java.lang.Object

└─ tud.iir.extraction.fact.FactValueComparator

### All Implemented Interfaces:

java.io.Serializable, java.util.Comparator

---

public class **FactValueComparator**  
extends java.lang.Object  
implements java.util.Comparator, java.io.Serializable

Sort facts by their trust.

**Author:**

David

---

## Constructors

### FactValueComparator

public **FactValueComparator**()

## Methods

### compare

public int **compare**([FactValue](#) fv1,  
                    [FactValue](#) fv2)

Highest trust first.

#### Parameters:

fv1 - FactValue1

fv2 - FactValue2

#### Returns:

0 or 1 depending on the trust.

## tud.iir.extraction.fact Class LiveFactExtractor

```
java.lang.Object
|
|--tud.iir.extraction.fact.LiveFactExtractor
```

---

```
public class LiveFactExtractor
extends java.lang.Object
```

The LiveFactExtractor manages fact extraction for entity names of unknown concepts. Only the names of the entities are known.

**Author:**  
David Urbansky

---

### Constructors

#### LiveFactExtractor

```
public LiveFactExtractor(java.lang.String entityName)
```

### Methods

#### extractFacts

```
public java.util.ArrayList extractFacts(int numberOfPages)
```

Extract facts for the entity name.

**Parameters:**

`numberOfPages` - The number of pages that are searched through for facts.

**Returns:**

An array of extracted facts.

---

#### extractFacts

```
public java.util.ArrayList extractFacts(java.lang.String url)
```

---

#### getEntityName

```
public java.lang.String getEntityName()
```

---

#### setEntityName

```
public void setEntityName(java.lang.String entityName)
```

---

## main

```
public static void main(java.lang.String[] args)
```

# tud.iir.extraction.fact

## Class NumericFactDistribution

```
java.lang.Object
└--tud.iir.extraction.fact.NumericFactDistribution
```

---

```
public class NumericFactDistribution
extends java.lang.Object
```

This class keeps track of the distribution of numeric facts.

**Author:**  
David Urbansky

---

## Constructors

### NumericFactDistribution

```
public NumericFactDistribution()
```

## Methods

### main

```
public static void main(java.lang.String[] args)
```

**Parameters:**  
args

---

### addNumber

```
public static void addNumber(java.lang.String key,
                             double number)
```

---

### getPowerDistributionFactor

```
public static double getPowerDistributionFactor(java.lang.String key,
                                                double number)
```

---

### getPowerDistributionFactor

```
public static double getPowerDistributionFactor(java.lang.String key,
                                                int power)
```

---

---

Package

**tud.iir.extraction.mio**



## tud.iir.extraction.mio

# Class AbstractMIOTypeExtractor

java.lang.Object

└- tud.iir.extraction.mio.AbstractMIOTypeExtractor

Direct Known Subclasses:

[AppletExtractor](#), [FlashExtractor](#), [HTML5CanvasExtractor](#), [QuicktimeExtractor](#),  
[SilverlightExtractor](#)

---

```
public abstract class AbstractMIOTypeExtractor
extends java.lang.Object
```

---

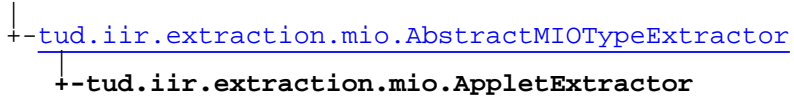
## Constructors

### AbstractMIOTypeExtractor

```
public AbstractMIOTypeExtractor()
```

## tud.iir.extraction.mio Class AppletExtractor

java.lang.Object



---

public class **AppletExtractor**  
extends [AbstractMIOTypeExtractor](#)

---

## Constructors

### AppletExtractor

public **AppletExtractor**()

## tud.iir.extraction.mio Class DedicatedPageDetector

```
java.lang.Object
└--tud.iir.extraction.mio.DedicatedPageDetector
```

---

```
public class DedicatedPageDetector
extends java.lang.Object
```

The DedicatedPageDetector calculate for a given MIOPage a TrustValue for being a DedicatedPage.

**Author:**  
Martin Werner

---

### Constructors

#### DedicatedPageDetector

```
public DedicatedPageDetector()
```

### Methods

#### calculateDedicatedPageTrust

```
public void calculateDedicatedPageTrust(MIOPage mioPage)
```

Calculate dedicated page trust.

**Parameters:**

`mioPage` - the mioPage

# tud.iir.extraction.mio

## Class EntityMIOExtractionThread

```

java.lang.Object
  |
  +- java.lang.Thread
        |
        +- tud.iir.extraction.mio.EntityMIOExtractionThread

```

All Implemented Interfaces:  
java.lang.Runnable

```

public class EntityMIOExtractionThread
extends java.lang.Thread

```

The EntityMIOExtractionThread extracts MIOs for one given entity. Therefore, extracting MIOs can be parallelized on the entity level.

**Author:**  
Martin Werner

## Constructors

### EntityMIOExtractionThread

```

public EntityMIOExtractionThread(java.lang.ThreadGroup threadGroup,
                                java.lang.String entityName,
                                Entity entity,
                                KnowledgeManager knowledgeManager)

```

Instantiates a new entity MIOExtractionThread.

**Parameters:**

threadGroup - the thread group  
entityName - the entityName  
entity - the entity  
knowledgeManager - the knowledge manager

## Methods

### run

```

public void run()

```

## tud.iir.extraction.mio Class FastMIODetector

```
java.lang.Object
└-- tud.iir.extraction.mio.FastMIODetector
```

---

```
public class FastMIODetector
extends java.lang.Object
```

The FastMIODetector simply analyze a MIOPageCandidate for pure MIO-Existence by some indicators.

**Author:**  
Martin Werner

---

### Constructors

#### FastMIODetector

```
public FastMIODetector()
```

Instantiates a new fast MIODetector.

### Methods

#### containsMIO

```
public boolean containsMIO(java.lang.String mioPageContent)
```

check if a MIO-Indicator is contained.

**Parameters:**

`mioPageContent` - the MIOPageContent

**Returns:**

true, if successful

## tud.iir.extraction.mio

### Class FlashExtractor

```
java.lang.Object
├── tud.iir.extraction.mio.AbstractMIOTypeExtractor
│   └── tud.iir.extraction.mio.FlashExtractor
```

---

```
public class FlashExtractor
extends AbstractMIOTypeExtractor
```

---

## Constructors

### FlashExtractor

```
public FlashExtractor()
```

## Methods

### main

```
public static void main(java.lang.String[] abc)
```

## tud.iir.extraction.mio

### Class HTML5CanvasExtractor

java.lang.Object

└- [tud.iir.extraction.mio.AbstractMIOTypeExtractor](#)  
└- tud.iir.extraction.mio.HTML5CanvasExtractor

---

public class HTML5CanvasExtractor  
extends [AbstractMIOTypeExtractor](#)

---

## Constructors

### HTML5CanvasExtractor

public HTML5CanvasExtractor()

# tud.iir.extraction.mio

## Class IFrameAnalyzer

java.lang.Object

└─ tud.iir.extraction.mio.IFrameAnalyzer

public class **IFrameAnalyzer**  
extends java.lang.Object

The IFrameAnalyzer analyzes a webPage for existing IFrames and checks if their targets contains MIOs.

**Author:**

Martin Werner

## Constructors

### IFrameAnalyzer

public **IFrameAnalyzer**([SearchWordMatcher](#) swMatcher)

Instantiates a new i frame analyzer.

**Parameters:**

swMatcher - the searchWordMatcher

## Methods

### getIframeMioPages

public java.util.List **getIframeMioPages**(java.lang.String parentPageContent,  
java.lang.String parentPageURL)

Gets the iframe mio pages.

**Parameters:**

parentPageContent - the parent page content

parentPageURL - the parent page URL

**Returns:**

the iframe mio pages



# tud.iir.extraction.mio

## Class InCoFiConfiguration

```
java.lang.Object
└--tud.iir.extraction.mio.InCoFiConfiguration
```

```
public class InCoFiConfiguration
extends java.lang.Object
```

The Class ConceptSearchVocabulary.

## Fields

### mobilephone

```
public transient java.lang.String mobilephone
```

The mobile phone.

### printer

```
public transient java.lang.String printer
```

The printer.

### headphone

```
public transient java.lang.String headphone
```

The headphone.

### movie

```
public transient java.lang.String movie
```

The movie.

### car

```
public transient java.lang.String car
```

The car.

### weakMIOS

```
public transient java.lang.String weakMIOS
```

The weak MIOs.

### resultCount

```
public transient int resultCount
```

(continued from last page)

The result count.

---

## searchEngine

```
public transient int searchEngine
```

The search engine.

---

## tempDirPath

```
public transient java.lang.String tempDirPath
```

The tempDirectoryPath.

---

## rolePageTrustLimit

```
public transient double rolePageTrustLimit
```

The RolePage Trust Limit

---

## rolePageRelevanceValue

```
public transient int rolePageRelevanceValue
```

The role page relevance value.

---

## analyzeSWFContent

```
public transient boolean analyzeSWFContent
```

Analyze SWFContent.

---

## limitLinkAnalyzing

```
public transient boolean limitLinkAnalyzing
```

Indicator for limiting the linkAnalyzing.

---

## mioTypes

```
public transient java.lang.String mioTypes
```

The relevant MIOTypes.

---

## redoWeak

```
public transient boolean redoWeak
```

---

## badWords

```
public transient java.lang.String badWords
```

The bad words.

---

(continued from last page)

---

## weakInteractionIndicators

```
public transient java.lang.String weakInteractionIndicators
```

The weak interaction indicators.

---

## strongInteractionIndicators

```
public transient java.lang.String strongInteractionIndicators
```

The strong interaction indicators.

---

## instance

```
public static tud.iir.extraction.mio.InCoFiConfiguration instance
```

The instance.

## Constructors

### InCoFiConfiguration

```
public InCoFiConfiguration()
```

## Methods

### getInstance

```
public static InCoFiConfiguration getInstance()
```

Gets the single instance of InCoFiConfiguration.

**Returns:**

single instance of InCoFiConfiguration

---

### getMIOTypes

```
public java.util.List getMIOTypes()
```

Gets the mIO types.

**Returns:**

the mIO types

---

### getBadWords

```
public java.util.List getBadWords()
```

Gets the bad words.

**Returns:**

the bad words

---

(continued from last page)

---

## getStrongInteractionIndicators

```
public java.util.List getStrongInteractionIndicators()
```

Gets the strong interaction indicators.

**Returns:**  
the strong interaction indicators

---

## getWeakInteractionIndicators

```
public java.util.List getWeakInteractionIndicators()
```

Gets the weak interaction indicators.

**Returns:**  
the weak interaction indicators

---

## getWeakMIOVocabulary

```
public java.util.List getWeakMIOVocabulary()
```

---

## getVocByConceptName

```
public java.util.List getVocByConceptName(java.lang.String conceptName)
```

Gets the searchVocabulary by concept name.

**Parameters:**  
conceptName - the concept name

**Returns:**  
the searchVocabulary by concept name

---

# tud.iir.extraction.mio

## Class LinkAnalyzer

```
java.lang.Object
└--tud.iir.extraction.mio.LinkAnalyzer
```

---

```
public class LinkAnalyzer
extends java.lang.Object
```

The LinkAnalyzer checks if some of the Links of MIOPageCandidates have targets with MIOs (simulates an indirect search)

**Author:**  
Martin Werner

---

## Constructors

### LinkAnalyzer

```
public LinkAnalyzer(SearchWordMatcher swMatcher,
                   Concept concept)
```

Instantiates a new LinkAnalyzer.

**Parameters:**

swMatcher - the SearchWordMatcher  
concept - the concept

## Methods

### getLinkedMioPages

```
public java.util.List getLinkedMioPages(java.lang.String parentPageContent,
                                         java.lang.String parentPageURL)
```

Gets the linked MIOpages.

**Parameters:**

parentPageContent - the parent page content  
parentPageURL - the parent pageURL

**Returns:**

the linked MIOPages

# tud.iir.extraction.mio

## Class MIO

```
java.lang.Object
└--tud.iir.extraction.mio.MIO
```

```
public class MIO
extends java.lang.Object
```

An interactive multimedia object.

**Author:**  
Martin Werner

## Constructors

### MIO

```
public MIO(java.lang.String mioType,
           java.lang.String directURL,
           java.lang.String findPageURL,
           Entity entity)
```

Instantiates a new MIO.

**Parameters:**

`mioType` - the MIOtype  
`directURL` - the directLinkURL  
`findPageURL` - the find page URL  
`entity` - the entity

## Methods

### initializeFeatures

```
public void initializeFeatures()
```

Initialize features.

### getTrust

```
public double getTrust()
```

Gets the trust.

**Returns:**  
the trust

### setTrust

```
public void setTrust(double trust)
```

Sets the trust.

**Parameters:**

(continued from last page)

---

trust - the new trust

---

## getFindPageURL

```
public java.lang.String getFindPageURL()
```

Gets the find page URL.

**Returns:**  
the find page URL

---

## setFindPageURL

```
public void setFindPageURL(java.lang.String findPageURL)
```

Sets the find page URL.

**Parameters:**  
findPageURL - the new find page URL

---

## getDirectURL

```
public java.lang.String getDirectURL()
```

Gets the direct URL.

**Returns:**  
the direct URL

---

## setDirectURL

```
public void setDirectURL(java.lang.String directURL)
```

Sets the direct url.

**Parameters:**  
directURL - the new direct url

---

## getEntity

```
public Entity getEntity()
```

Gets the entity.

**Returns:**  
the entity

---

## setEntity

```
public void setEntity(Entity entity)
```

Sets the entity.

**Parameters:**  
entity - the new entity

---

(continued from last page)

---

## getInteractivityGrade

```
public java.lang.String getInteractivityGrade()
```

Gets the interactivity grade.

**Returns:**  
the interactivity grade

---

## setInteractivityGrade

```
public void setInteractivityGrade(java.lang.String interactivityGrade)
```

Sets the interactivity grade.

**Parameters:**  
interactivityGrade - the new interactivity grade

---

## isDedicatedPage

```
public boolean isDedicatedPage()
```

Checks if is dedicated page.

**Returns:**  
true, if is dedicated page

---

## setDedicatedPage

```
public void setDedicatedPage(boolean isDedicatedPage)
```

Sets the dedicated page.

**Parameters:**  
isDedicatedPage - the new dedicated page

---

## getMIOType

```
public java.lang.String getMIOType()
```

Gets the type.

**Returns:**  
the type

---

## setMIOType

```
public void setMIOType(java.lang.String type)
```

Sets the type.

**Parameters:**  
type - the new type

---

## getFileName

```
public java.lang.String getFileName()
```

---



(continued from last page)

Gets the file name.

**Returns:**  
the file name

---

## setFileName

```
public void setFileName(java.lang.String fileName)
```

Sets the file name.

**Parameters:**  
fileName - the new file name

---

## setFeature

```
public void setFeature(java.lang.String name,  
double value)
```

Sets the feature.

**Parameters:**  
name - the name  
value - the value

---

## getFeature

```
public double getFeature(java.lang.String name)
```

Gets the feature.

**Parameters:**  
name - the name

**Returns:**  
the feature

---

## getFeatures

```
public java.util.Map getFeatures()
```

Gets the features.

**Returns:**  
the features

---

## getFileSize

```
public double getFileSize()
```

Gets the file size.

**Returns:**  
the file size

---

## setFileSize

```
public void setFileSize(double fileSize)
```

---

---

(continued from last page)

Sets the file size.

**Parameters:**

`fileSize` - the new file size

---

## setFeatures

```
public void setFeatures(java.util.Map features)
```

Sets the features.

**Parameters:**

`features` - the features

---

## getAltText

```
public java.lang.String getAltText()
```

---

## setAltText

```
public void setAltText(java.lang.String altText)
```

---

## getPreviousHeadlines

```
public java.lang.String getPreviousHeadlines()
```

---

## setPreviousHeadlines

```
public void setPreviousHeadlines(java.lang.String previousHeadlines)
```

---

## getSurroundingText

```
public java.lang.String getSurroundingText()
```

---

## setSurroundingText

```
public void setSurroundingText(java.lang.String surroundingText)
```

---

## tud.iir.extraction.mio Class MIOComparator

java.lang.Object

└─ tud.iir.extraction.mio.MIOComparator

### All Implemented Interfaces:

java.io.Serializable, java.util.Comparator

---

```
public class MIOComparator
extends java.lang.Object
implements java.util.Comparator, java.io.Serializable
```

---

## Constructors

### MIOComparator

```
public MIOComparator()
```

## Methods

### compare

```
public int compare(java.lang.Object obj1,
                   java.lang.Object obj2)
```

# tud.iir.extraction.mio

## Class MIOContextAnalyzer

java.lang.Object

└─ tud.iir.extraction.mio.MIOContextAnalyzer

public class **MIOContextAnalyzer**  
extends java.lang.Object

The Class MIOContextAnalyzer analyze the context and sets the features.

## Constructors

### MIOContextAnalyzer

```
public MIOContextAnalyzer(Entity entity,  
                           MIOPage mioPage)
```

Instantiates a new mioContextAnalyzer.

**Parameters:**

entity - the entity

mioPage - the mioPage

## Methods

### setFeatures

```
public void setFeatures(MIO mio)
```

Sets the features.

**Parameters:**

mio - the new features

### extractXMLContent

```
public static java.lang.String extractXMLContent(java.lang.String xmlFileURL)
```

Extract the content of an XML-File.

**Parameters:**

xmlFileURL - the XML-File-URL

**Returns:**

the complete Content(incl. tags) as String

# tud.iir.extraction.mio

## Class MIOExtractionProcess

```
java.lang.Object
├-- java.lang.Thread
│   └-- tud.iir.extraction.mio.MIOExtractionProcess
```

All Implemented Interfaces:  
java.lang.Runnable

```
public class MIOExtractionProcess
extends java.lang.Thread
```

## Constructors

### MIOExtractionProcess

```
public MIOExtractionProcess()
```

## Methods

### run

```
public void run()
```

### stopExtraction

```
public boolean stopExtraction()
```

Stop extraction.

**Returns:**

true, if successful

### isBenchmark

```
public boolean isBenchmark()
```

Checks if is benchmark.

**Returns:**

true, if is benchmark

### setBenchmark

```
public void setBenchmark(boolean benchmark)
```

Sets the benchmark.

(continued from last page)

**Parameters:**

benchmark - the new benchmark

# tud.iir.extraction.mio

## Class MIOExtractor

```
java.lang.Object
├── tud.iir.extraction.Extractor
│   └── tud.iir.extraction.mio.MIOExtractor
```

public final class **MIOExtractor**  
extends [Extractor](#)

## Methods

### getInstance

public static [MIOExtractor](#) **getInstance()**

Gets the single instance of MIOExtractor.

**Returns:**  
single instance of MIOExtractor

### startExtraction

public void **startExtraction()**

Start extraction of MIOs for entities that are fetched from the knowledge base. Continue from last extraction.

### startExtraction

public void **startExtraction**(boolean continueFromLastExtraction)

Start extraction.

**Parameters:**  
continueFromLastExtraction - the continue from last extraction

### isURLallowed

public boolean **isURLallowed**(java.lang.String url)

Check if URL is allowed.

**Parameters:**  
url - the URL

**Returns:**  
true, if the URL allowed

(continued from last page)

## main

```
public static void main(java.lang.String[] abc)
```

The main method.

### Parameters:

abc - the arguments



# tud.iir.extraction.mio

## Class MIOInteractivityAnalyzer

java.lang.Object

└─ tud.iir.extraction.mio.MIOInteractivityAnalyzer

---

```
public class MIOInteractivityAnalyzer
extends java.lang.Object
```

---

## Constructors

### MIOInteractivityAnalyzer

```
public MIOInteractivityAnalyzer()
```

Instantiates a new mIO interactivity analyzer.

## Methods

### setInteractivityGrade

```
public void setInteractivityGrade(MIO mio,
MIOPage mioPage)
```

Sets the interactivity grade. If textual content exists, mostly the MIO is strong. If the fileSize is bigger than 2097152 Byte (=2MB), mostly the MIO is a video and thats why weak.

#### Parameters:

mio - the mio

mioPage - the mioPage

# tud.iir.extraction.mio

## Class MIOPage

```
java.lang.Object
|
+--tud.iir.extraction.mio.MIOPage
```

---

```
public class MIOPage
extends java.lang.Object
```

An webpage which contains mio(s).

**Author:**  
Martin Werner

---

## Constructors

### MIOPage

```
public MIOPage(java.lang.String url)
```

Instantiates a new mIO page.

**Parameters:**  
url - the URL

---

### MIOPage

```
public MIOPage(java.lang.String url,
               org.w3c.dom.Document webDocument)
```

Instantiates a new mIO page.

**Parameters:**  
url - the url  
webDocument - the web document

---

## Methods

### getUrl

```
public java.lang.String getUrl()
```

Gets the url.

**Returns:**  
the url

---

### setUrl

```
public void setUrl(java.lang.String url)
```

Sets the url.

**Parameters:**  
url - the new url

---

---

## getHostname

```
public java.lang.String getHostname()
```

Gets the hostname.

**Returns:**  
the hostname

---

## isIFrameSource

```
public boolean isIFrameSource()
```

Checks if is i frame source.

**Returns:**  
true, if is i frame source

---

## setIFrameSource

```
public void setIFrameSource(boolean isIFrameSource)
```

Sets the i frame source.

**Parameters:**  
`isIFrameSource` - the new i frame source

---

## getContentAsString

```
public java.lang.String getContentAsString()
```

Gets the content.

**Returns:**  
the content

---

## getLinkName

```
public java.lang.String getLinkName()
```

Gets the link name.

**Returns:**  
the link name

---

## setLinkName

```
public void setLinkName(java.lang.String linkName)
```

Sets the link name.

**Parameters:**  
`linkName` - the new link name

---

## getLinkParentPage

```
public java.lang.String getLinkParentPage()
```

(continued from last page)

Gets the link parent page.

**Returns:**  
the link parent page

---

## setLinkParentPage

```
public void setLinkParentPage(java.lang.String linkParentPage)
```

Sets the link parent page.

**Parameters:**  
linkParentPage - the new link parent page

---

## isLinkedPage

```
public boolean isLinkedPage()
```

Checks if is linked page.

**Returns:**  
true, if is linked page

---

## setLinkedPage

```
public void setLinkedPage(boolean isLinkedPage)
```

Sets the linked page.

**Parameters:**  
isLinkedPage - the new linked page

---

## getLinkTitle

```
public java.lang.String getLinkTitle()
```

Gets the link title.

**Returns:**  
the link title

---

## setLinkTitle

```
public void setLinkTitle(java.lang.String linkTitle)
```

Sets the link title.

**Parameters:**  
linkTitle - the new link title

---

## getDedicatedPageTrust

```
public double getDedicatedPageTrust()
```

Gets the dedicated page trust.

**Returns:**  
the dedicated page trust

---

---

## setDedicatedPageTrust

```
public void setDedicatedPageTrust(double dedicatedPageTrust)
```

Sets the dedicated page trust.

**Parameters:**

dedicatedPageTrust - the new dedicated page trust

---

## getIframeParentPage

```
public java.lang.String getIframeParentPage()
```

Gets the iframe parent page.

**Returns:**

the iframe parent page

---

## setIframeParentPage

```
public void setIframeParentPage(java.lang.String iframeParentPage)
```

Sets the iframe parent page.

**Parameters:**

iframeParentPage - the new iframe parent page

---

## setIframeParentPageTitle

```
public void setIframeParentPageTitle(java.lang.String iframeParentPageTitle)
```

Sets the iframe parent page title.

**Parameters:**

iframeParentPageTitle - the new iframe parent page title

---

## getIframeParentPageTitle

```
public java.lang.String getIframeParentPageTitle()
```

Gets the iframe parent page title.

**Returns:**

the iframe parent page title

---

## getTitle

```
public java.lang.String getTitle()
```

Gets the title.

**Returns:**

the title

---

## setTitle

```
public void setTitle(java.lang.String title)
```

---

(continued from last page)

Sets the title.

**Parameters:**

title - the new title

---

## getWebDocument

```
public org.w3c.dom.Document getWebDocument()
```

Gets the web document.

**Returns:**

the web document

## tud.iir.extraction.mio

# Class MIOPageCandidateAnalyzer

java.lang.Object

└─tud.iir.extraction.mio.MIOPageCandidateAnalyzer

---

public class **MIOPageCandidateAnalyzer**  
extends java.lang.Object

The PageAnalyzer analyzes MIOPageCandidates for MIO-Existence. Also some links and IFRAMEs are analyzed.

**Author:**  
Martin Werner

---

## Constructors

### MIOPageCandidateAnalyzer

public **MIOPageCandidateAnalyzer**(java.util.List mioPageCandidates)

Instantiates a new PageAnalyzer.

**Parameters:**

mioPageCandidates - the mIO page candidates

## Methods

### identifyMIOPages

public final java.util.List **identifyMIOPages**([Entity](#) entity)

This central method identifies entity-relevant MIOPages.

**Parameters:**

entity - the entity

**Returns:**

the identified mioPages as list

## tud.iir.extraction.mio

# Class MIOPageRetriever

```
java.lang.Object
└--tud.iir.extraction.mio.MIOPageRetriever
```

---

```
public class MIOPageRetriever
extends java.lang.Object
```

The MIOPageRetriever finds pages from the web that have a relative high probability of containing relevant MIO(s) for a given entity.

**Author:**  
Martin Werner

---

## Constructors

### MIOPageRetriever

```
public MIOPageRetriever()
```

## Methods

### retrieveMIOPages

```
public java.util.List retrieveMIOPages(Entity entity,
boolean weakFlag)
```

Retrieve MIOs.

**Parameters:**

`entity` - the entity

**Returns:**

the list



## tud.iir.extraction.mio

# Class MIOQueryFactory

java.lang.Object

└-tud.iir.extraction.mio.MIOQueryFactory

---

public class **MIOQueryFactory**  
extends java.lang.Object

The MIOQueryFactory creates a List of specific SearchQueries for a given entity and concept

**Author:**  
Martin Werner

---

## Methods

### generateSearchQueries

public java.util.List **generateSearchQueries**()

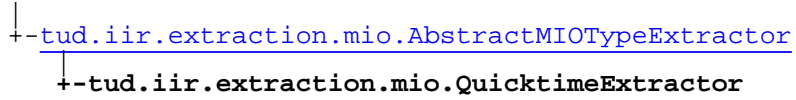
Generate search queries.

**Returns:**  
the list

## tud.iir.extraction.mio

### Class QuicktimeExtractor

java.lang.Object



public class QuicktimeExtractor  
extends [AbstractMIOTypeExtractor](#)

---

## Constructors

### QuicktimeExtractor

public QuicktimeExtractor()

## tud.iir.extraction.mio Class RelevanceCalculator

```
java.lang.Object
├--tud.iir.extraction.mio.RelevanceCalculator
```

---

```
public final class RelevanceCalculator
extends java.lang.Object
```

---

### Methods

#### calcStringRelevance

```
public static double calcStringRelevance(java.lang.String inputString,
    Entity entity)
```

Calculates string relevance.

**Parameters:**

inputString - the input string  
entity - the entity

**Returns:**

the double

---

#### calcStringRelevance

```
public static double calcStringRelevance(java.lang.String inputString,
    java.lang.String entityName)
```

Calculates the relevance of a string by checking how many terms or morphs of the entityName are included in the string. A special role play words like d500 or x500i. Returns: a value from 0 to 1

**Parameters:**

inputString - the string  
entityName - the entity name

**Returns:**

the double

---

# tud.iir.extraction.mio

## Class RolePage

```
java.lang.Object
└--tud.iir.extraction.mio.RolePage
```

```
public class RolePage
extends java.lang.Object
```

A RolePage is a Page which has a central role within a concept, e.g. www.gsmarena.com for concept mobilePhone

**Author:**  
Martin Werner

## Constructors

### RolePage

```
public RolePage(java.lang.String hostname,
                int conceptID)
```

Instantiates a new rolePage.

**Parameters:**

hostname - the hostname  
conceptID - the concept id

### RolePage

```
public RolePage(java.lang.String hostname,
                int count,
                int conceptID)
```

Instantiates a new rolePage (especially for loading from database).

**Parameters:**

hostname - the hostname  
count - the count  
conceptID - the concept id

## Methods

### incrementCount

```
public void incrementCount()
```

Calc count.

### getHostname

```
public java.lang.String getHostname()
```

Gets the hostname.

(continued from last page)

**Returns:**  
the hostname

---

## setHostname

```
public void setHostname(java.lang.String hostname)
```

Sets the hostname.

**Parameters:**  
hostname - the new hostname

---

## getCount

```
public int getCount()
```

Gets the count.

**Returns:**  
the count

---

## setCount

```
public void setCount(int count)
```

Sets the count.

**Parameters:**  
count - the new count

---

## getID

```
public int getID()
```

Gets the id.

**Returns:**  
the id

---

## setID

```
public void setID(int id)
```

Sets the id.

**Parameters:**  
id - the new id

---

## getConceptID

```
public int getConceptID()
```

Gets the concept id.

**Returns:**  
the concept id

(continued from last page)

## setConceptID

```
public void setConceptID(int conceptID)
```

Sets the concept id.

### Parameters:

`conceptID` - the new concept id

# tud.iir.extraction.mio

## Class RolePageDatabase

java.lang.Object

└─tud.iir.extraction.mio.RolePageDatabase

public class **RolePageDatabase**  
extends java.lang.Object

## Methods

### loadNotUsedRolePagesForEntity

public java.util.List **loadNotUsedRolePagesForEntity**([Entity](#) entity)

Load all rolePages, that were not already used for the specific entity.

**Parameters:**

entity - the entity

**Returns:**

the array list

### loadUsedRolePageIDsForEntity

public java.util.List **loadUsedRolePageIDsForEntity**([Entity](#) entity)

Load all IDs of rolePages that where already used for a specific entity.

**Parameters:**

entity - the entity

**Returns:**

the array list

### loadAllRolePagesForConcept

public java.util.List **loadAllRolePagesForConcept**([Concept](#) concept)

Load all RolePages which are associated with a specific concept.

**Parameters:**

concept - the concept

**Returns:**

the array list

### insertRolePage

public void **insertRolePage**([RolePage](#) rolePage)

Adds the rolePage to database.

(continued from last page)

**Parameters:**rolePage - the rolePage

---

## insertRolePageUsage

```
public void insertRolePageUsage(RolePage rolePage,  
    Entity entity)
```

Insert rolePage usage.

**Parameters:**

rolePage - the rolePage

entity - the entity

---

## updateRolePage

```
public void updateRolePage(RolePage rolePage)
```

Update a rolePage in database.

**Parameters:**rolePage - the rolePage

---

## removeUnrelevantRolePages

```
public void removeUnrelevantRolePages(int minCount)
```

Remove all rolePages from database that don't fit a concrete minimalCount.

**Parameters:**minCount - the minimalCount

---



## tud.iir.extraction.mio

# Class RolePageDetector

java.lang.Object

└-tud.iir.extraction.mio.RolePageDetector

---

public class **RolePageDetector**  
extends java.lang.Object

Detects RolePages

**Author:**

Martin Werner

---

## Methods

### detectRolePages

public void **detectRolePages**(java.util.Set sortedMIOS)

Detect role pages.

**Parameters:**

sortedMIOS - the sorted MIOs

# tud.iir.extraction.mio

## Class SearchAgent

```
java.lang.Object
└-- tud.iir.extraction.mio.SearchAgent
```

---

```
public class SearchAgent
extends java.lang.Object
```

The SearchAgent uses given queries to initiate a search at a searchEngine.

**Author:**  
Martin Werner

---

## Constructors

### SearchAgent

```
public SearchAgent()
```

Instantiates a new search agent.

## Methods

### initiateSearch

```
public java.util.List initiateSearch(java.util.List searchQueries)
```

Initiate search.

**Parameters:**

`searchQueries` - the search queries

**Returns:**

the list

# tud.iir.extraction.mio

## Class SearchWordMatcher

```
java.lang.Object
└--tud.iir.extraction.mio.SearchWordMatcher
```

```
public class SearchWordMatcher
extends java.lang.Object
```

The SearchWordMatcher checks if and how deep a given String contains an EntityName or a morpheme of it.

**Author:**  
Martin Werner

## Constructors

### SearchWordMatcher

```
public SearchWordMatcher(java.lang.String searchWords)
```

By instantiating a list of words is generated out of the given searchwords (entityName).

**Parameters:**

searchWords - the search words

## Methods

### getNumberOfSearchWordMatches

```
public int getNumberOfSearchWordMatches(java.lang.String src)
```

Check how deep a searchword or a kind of morphing is contained in the string ("samsung" vs. "samsung S8500")

**Parameters:**

src - the src

**Returns:**

the number of search word matches

### getNumberOfSearchWordMatches

```
public int getNumberOfSearchWordMatches(java.lang.String src,
boolean withoutSpecialWords,
java.lang.String searchWords)
```

Gets the number of search word matches.

**Parameters:**

src - the src

withoutSpecialWords - the without special words

searchWords - the search words

**Returns:**

(continued from last page)

the number of search word matches

---

## containsSearchWordOrMorphs

```
public boolean containsSearchWordOrMorphs(java.lang.String src)
```

Check if a searchword or a kind of morphing is contained in the string. If the name of entity consists of more words, than the half of them must minimally be contained in the given string.

### Parameters:

`src` - the src

### Returns:

true, if successful

---

## main

```
public static void main(java.lang.String[] args)
```

The main method.

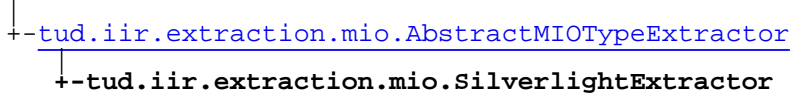
### Parameters:

`args` - the arguments

## tud.iir.extraction.mio

### Class SilverlightExtractor

java.lang.Object



---

```
public class SilverlightExtractor
extends AbstractMIOTypeExtractor
```

---

## Constructors

### SilverlightExtractor

```
public SilverlightExtractor()
```

# tud.iir.extraction.mio

## Class SWFContentAnalyzer

```
java.lang.Object
  |
  +-SWFTagTypesImpl
    |
    +-tud.iir.extraction.mio.SWFContentAnalyzer
```

```
public class SWFContentAnalyzer
extends SWFTagTypesImpl
```

Parse a Flash movie and extract all the text in Text symbols A "pipeline" is set up: SWFReader-->TagParser-->SWFContentAnalyzer SWFReader reads the input SWF file and separates out the header and the tags. The separated contents are passed to TagParser which parses out the individual tag types and passes them to SWFContentAnalyzer. SWFContentAnalyzer extends SWFTagTypesImpl and overrides some methods.

**Author:**

Martin Werner

## Constructors

### SWFContentAnalyzer

```
public SWFContentAnalyzer()
```

Instantiates a new MIOContentAnalyzer.

## Methods

### tagDefineFontInfo

```
public void tagDefineFontInfo(int fontId,
    java.lang.String fontName,
    int flags,
    int[] codes)
throws java.io.IOException
```

SWFTagTypes interface Save the Text Font character code info.

**Parameters:**

fontId - the font id  
fontName - the font name  
flags - the flags  
codes - the codes

**Throws:**

IOException - Signals that an I/O exception has occurred.

(continued from last page)

## tagDefineFont2

```
public SWFVectors tagDefineFont2(int tagID,  
    int flags,  
    java.lang.String name,  
    int numGlyphs,  
    int ascent,  
    int descent,  
    int leading,  
    int[] codes,  
    int[] advances,  
    Rect[] bounds,  
    int[] kernCodes1,  
    int[] kernCodes2,  
    int[] kernAdjustments)  
throws java.io.IOException
```

SWFTagTypes interface Save the character code info.

### Parameters:

tagID - the id  
flags - the flags  
name - the name  
numGlyphs - the num glyphs  
ascent - the ascent  
descent - the descent  
leading - the leading  
codes - the codes  
advances - the advances  
bounds - the bounds  
kernCodes1 - the kern codes1  
kernCodes2 - the kern codes2  
kernAdjustments - the kern adjustments

### Returns:

the SWF vectors

### Throws:

IOException - Signals that an I/O exception has occurred.

---

## tagDefineTextField

```
public void tagDefineTextField(int fieldId,  
    java.lang.String fieldName,  
    java.lang.String initialText,  
    Rect boundary,  
    int flags,  
    AlphaColor textColor,  
    int alignment,  
    int fontId,  
    int fontSize,  
    int charLimit,  
    int leftMargin,  
    int rightMargin,  
    int indentation,  
    int lineSpacing)  
throws java.io.IOException
```

SWFTagTypes interface Dump any initial text in the field.

### Parameters:

fieldId - the field id  
fieldName - the field name

(continued from last page)

initialText - the initial text  
 boundary - the boundary  
 flags - the flags  
 textColor - the text color  
 alignment - the alignment  
 fontId - the font id  
 fontSize - the font size  
 charLimit - the char limit  
 leftMargin - the left margin  
 rightMargin - the right margin  
 indentation - the indentation  
 lineSpacing - the line spacing

**Throws:**

`IOException` - Signals that an I/O exception has occurred.

## tagDefineText

```
public SWFText tagDefineText(int someId,
    Rect bounds,
    Matrix matrix)
    throws java.io.IOException
```

SWFTagTypes interface.

**Parameters:**

someId - the some id  
 bounds - the bounds  
 matrix - the matrix

**Returns:**

the SWF text

**Throws:**

`IOException` - Signals that an I/O exception has occurred.

## analyzeContentAndSetFeatures

```
public void analyzeContentAndSetFeatures(MIO mio,
    Entity entity)
```

This is the central method of this class and allows to completely analyze the content of a given SWF-MIO and add some relevant parameter and features to that MIO.

**Parameters:**

mio - the SWF-MIO  
 entity - the entity

## main

```
public static void main(java.lang.String[] args)
    throws java.io.IOException
```

The main method.

**Parameters:**

args - the arguments

**Throws:**

`IOException` - Signals that an I/O exception has occurred.



## tud.iir.extraction.mio

# Class SWFContentAnalyzer.TextDumper

java.lang.Object

└--tud.iir.extraction.mio.SWFContentAnalyzer.TextDumper

---

public class **SWFContentAnalyzer.TextDumper**  
extends java.lang.Object

---

## Constructors

### SWFContentAnalyzer.TextDumper

public **SWFContentAnalyzer.TextDumper**()

## Methods

### font

public void **font**(int fontId,  
int textHeight)

---

### setY

public void **setY**(int yvar)

---

### text

public void **text**(int[] glyphIndices,  
int[] glyphAdvances)

---

### color

public void **color**(Color color)

---

### setX

public void **setX**(int xvar)

---

(continued from last page)

## done

```
public void done()
```

## tud.iir.extraction.mio

# Class UniversalMIOExtractor

```
java.lang.Object
└-- tud.iir.extraction.mio.UniversalMIOExtractor
```

---

```
public class UniversalMIOExtractor
extends java.lang.Object
```

The Class UniversalMIOExtractor is a context-based MIO-Extractor.

---

## Constructors

### UniversalMIOExtractor

```
public UniversalMIOExtractor(Entity entity)
```

## Methods

### analyzeMIOPages

```
public java.util.List analyzeMIOPages(java.util.List mioPages)
```

---

Package

**tud.iir.extraction.object**

# tud.iir.extraction.object

## Class ObjectExtractor

java.lang.Object

└─ tud.iir.extraction.object.ObjectExtractor

All Implemented Interfaces:

[CrawlerCallback](#)

---

```
public class ObjectExtractor
extends java.lang.Object
implements CrawlerCallback
```

---

## Methods

### getInstance

```
public static ObjectExtractor getInstance()
```

Get the instance of the ObjectExtractor, which itself is singleton.

Returns:

The ObjectExtractor instance.

---

### loadObjectDescription

```
public void loadObjectDescription(boolean created)
```

---

### createTemplate

```
public void createTemplate()
```

---

### startCrawl

```
public void startCrawl()
```

---

### crawlerCallback

```
public void crawlerCallback(org.w3c.dom.Document document)
```

---

### applyExtractionTemplate

```
public void applyExtractionTemplate(org.w3c.dom.Document document)
```

(continued from last page)

---

## main

```
public static void main(java.lang.String[] args)
```

---

Package  
**tud.iir.extraction.qa**

## tud.iir.extraction.qa

# Class QAExtractionProcess

```
java.lang.Object
├-- java.lang.Thread
│   └-- tud.iir.extraction.qa.QAExtractionProcess
```

All Implemented Interfaces:  
java.lang.Runnable

---

```
public class QAExtractionProcess
extends java.lang.Thread
```

The QA extraction process.

**Author:**  
David Urbansky

---

## Constructors

### QAExtractionProcess

```
public QAExtractionProcess()
```

## Methods

### run

```
public void run()
```

---

### stopExtraction

```
public boolean stopExtraction()
```



# tud.iir.extraction.qa

## Class QAExtractionThread

```
java.lang.Object
├── java.lang.Thread
│   └── tud.iir.extraction.qa.QAExtractionThread
```

All Implemented Interfaces:  
java.lang.Runnable

```
public class QAExtractionThread
extends java.lang.Thread
```

## Constructors

### QAExtractionThread

```
public QAExtractionThread(java.lang.ThreadGroup threadGroup,
                           java.lang.String name,
                           QASite qaSite)
```

## Methods

### run

```
public void run()
```

### getPa

```
public PageAnalyzer getPa()
```

### setPa

```
public void setPa(PageAnalyzer pa)
```

### getCrawler

```
public Crawler getCrawler()
```

(continued from last page)

## setCrawler

```
public void setCrawler(Crawler crawler)
```

## tud.iir.extraction.qa Class QAExtractor

```
java.lang.Object
├── tud.iir.extraction.Extractor
│   └── tud.iir.extraction.qa.QAExtractor
```

---

```
public class QAExtractor
    extends Extractor
```

The main class for the Q/A extraction. (QUAX) QUAX knows a set of Q/A pages with information about question xPaths and answer xPaths. QUAX performs a focused crawl over the Q/A pages and remembers URLs of visited pages also over extraction session. New Q/As are extracted and written in the database.

**Author:**

David Urbansky

---

## Methods

### getInstance

```
public static QAExtractor getInstance()
```

Get the instance of the QAExtractor, which itself is singleton.

**Returns:**

The QAExtractor instance.

---

### setAnswerClassifier

```
public void setAnswerClassifier(int type)
```

---

### startExtraction

```
public void startExtraction()
```

The Q/A extraction is a bootstrapped process with two steps alternately performed in a loop: 1: use a seed query to retrieve urls with question and answers 2: perform a focused crawling on each retrieved url to increase the Q/A set

---

### startExtraction

```
public void startExtraction(boolean continueExtraction)
```

---

### extractFAQ

```
public java.util.ArrayList extractFAQ(java.lang.String url)
```

Analyze page for FAQ and extract QA tuples if possible.

**Parameters:**

---

(continued from last page)

url - The url to analyze.

**Returns:**

A set of QA tuples if an FAQ was found.

---

## addQA

```
public void addQA(QA qa)
```

Add a QA tuple and save them if they are over a certain number. This method is called by QAExtractionThreads and thus must be synchronized.

**Parameters:**

qa - The QA tuple to add.

---

## detectAnswer

```
public java.lang.String[] detectAnswer(java.lang.String question,  
    java.util.LinkedHashSet questionXPath)
```

Detect an answer without knowing its xPath. Build a candidate set, detect features and use a learned classifier to rank the candidates.

**Parameters:**

question  
pa

**Returns:**

A two entry long string array with the answer and its XPath.

---

## filterAnswerCandidates

```
public java.util.LinkedHashSet filterAnswerCandidates(java.util.LinkedHashSet  
questionXPaths,  
    java.util.LinkedHashSet answerCandidates)
```

Filter out candidates that point to the same or a parent xPath of the question.

**Returns:**

A filtered set of candidate answers.

---

## getAnswerFeatures

```
public AnswerFeatures getAnswerFeatures(java.lang.String question,  
    java.lang.String htmlAnswer)
```

Get features for the given answer.

**Returns:**

---

## runQAFromOfflineTestset

```
public void runQAFromOfflineTestset()
```

---

(continued from last page)

## getPa

```
public PageAnalyzer getPa()
```

---

## setPa

```
public void setPa(PageAnalyzer pa)
```

---

## main

```
public static void main(java.lang.String[] arguments)
```

# tud.iir.extraction.qa

## Class QASite

java.lang.Object

└--tud.iir.extraction.qa.QASite

All Implemented Interfaces:

java.io.Serializable

---

public class **QASite**  
extends java.lang.Object  
implements java.io.Serializable

---

## Fields

### FAQ

public static int **FAQ**

---

### QA\_SITE

public static int **QA\_SITE**

---

## Constructors

### QASite

public **QASite**(java.util.HashMap siteInformation)

---

## Methods

### getName

public java.lang.String **getName**()

---

### setName

public void **setName**(java.lang.String name)

---

(continued from last page)

## getType

```
public int getType()
```

---

## setType

```
public void setType(java.lang.String type)
```

---

## getMaximumURLs

```
public int getMaximumURLs()
```

---

## setMaximumURLs

```
public void setMaximumURLs(java.lang.Object maximumURLs)
```

---

## getEntryURL

```
public java.lang.String getEntryURL()
```

---

## setEntryURL

```
public void setEntryURL(java.lang.String entryURL)
```

---

## getQuestionXPath

```
public java.lang.String getQuestionXPath()
```

---

## setQuestionXPath

```
public void setQuestionXPath(java.lang.String questionXPath)
```

---

## getBestAnswerXPath

```
public java.lang.String getBestAnswerXPath()
```

---

## setBestAnswerXPath

```
public void setBestAnswerXPath(java.lang.String bestAnswerXPath)
```

---

---

## getAllAnswersXPath

```
public java.lang.String getAllAnswersXPath()
```

---

## setAllAnswersXPath

```
public void setAllAnswersXPath(java.lang.String allAnswersXPath)
```

---

## getAnswerPrefix

```
public java.lang.String getAnswerPrefix()
```

---

## setAnswerPrefix

```
public void setAnswerPrefix(java.lang.String answerPrefix)
```

---

## getAnswerSuffix

```
public java.lang.String getAnswerSuffix()
```

---

## setAnswerSuffix

```
public void setAnswerSuffix(java.lang.String answerSuffix)
```

---

## getURLStackSize

```
public int getURLStackSize()
```

---

## getURLFromStack

```
public QAUr1 getURLFromStack()
```

Try to get green prefix urls (pages with Q/As) first. If none of these is available try to get yellow prefix urls (urls that directly point to Q/A pages). If none of these is available, take any url.

**Returns:**

---

## addURLToStack

```
public void addURLToStack(QAUr1 url)
```



---

## removeURLFromStack

```
public void removeURLFromStack(QAUrl url)
```

---

## urlsAvailable

```
public boolean urlsAvailable()
```

---

## updatePositivePrefixes

```
public void updatePositivePrefixes(QAUrl url)
```

Update prefixes only if url is a page where at least a question was extracted.

### Parameters:

url - The url object.

---

## updateNegativePrefix

```
public void updateNegativePrefix(QAUrl url)
```

---

## getGreenPrefix

```
public java.lang.String getGreenPrefix()
```

---

## setGreenPrefix

```
public void setGreenPrefix(java.lang.String greenPrefix)
```

---

## greenPrefixCreated

```
public boolean greenPrefixCreated()
```

---

## setGreenPrefixCreated

```
public void setGreenPrefixCreated(boolean greenPrefixCreated)
```

---

## getGreenUrlDepth

```
public int getGreenUrlDepth()
```

---

## setGreenUrlDepth

```
public void setGreenUrlDepth(int greenUrlDepth)
```

---

---

## hasVoted

```
public boolean hasVoted()
```

---

---

## setVoted

```
public void setVoted()
```

---

---

## addQuestionHash

```
public boolean addQuestionHash(int questionHash)
```

Add the hash of a question. Return true if hash existed already, else false.

**Parameters:**

questionHash - The hash of the question.

**Returns:**

True if the question was extracted on the site already, false otherwise.

---

---

## getQuestionHashes

```
public java.util.TreeSet getQuestionHashes()
```

---

---

## setQuestionHashes

```
public void setQuestionHashes(java.util.TreeSet questionHashes)
```

---

---

## toString

```
public java.lang.String toString()
```

---

## tud.iir.extraction.qa Class QASites

```
java.lang.Object
├-- java.util.AbstractCollection
│   ├── java.util.AbstractList
│   │   ├── java.util.ArrayList
│   │   └-- tud.iir.extraction.qa.QASites
```

### All Implemented Interfaces:

java.io.Serializable, java.util.Collection, java.util.List, java.io.Serializable, java.lang.Cloneable,  
java.util.RandomAccess, java.util.List

```
public class QASites
extends java.util.ArrayList
implements java.util.List, java.util.RandomAccess, java.lang.Cloneable,
java.io.Serializable, java.util.List, java.util.Collection, java.io.Serializable
```

## Constructors

### QASites

```
public QASites()
```

## Methods

### getTotalURLStackSize

```
public int getTotalURLStackSize()
```

### serialize

```
public void serialize()
```

Serialize state of QASite extraction to resume later on.

## tud.iir.extraction.qa

### Class QAUrl

java.lang.Object

└-- tud.iir.extraction.qa.QAUrl

All Implemented Interfaces:

java.io.Serializable

---

```
public class QAUrl
    extends java.lang.Object
    implements java.io.Serializable
```

---

## Fields

### UNKNOWN

```
public static java.lang.String UNKNOWN
```

---

### GREEN

```
public static java.lang.String GREEN
```

---

### YELLOW

```
public static java.lang.String YELLOW
```

---

### NON\_RED

```
public static java.lang.String NON_RED
```

---

## Constructors

### QAUrl

```
public QAUrl(java.lang.String url,
             java.lang.String parentURL)
```

---

## Methods

(continued from last page)

## getUrl

```
public java.lang.String getUrl()
```

---

## setUrl

```
public void setUrl(java.lang.String url)
```

---

## getParentURL

```
public java.lang.String getParentURL()
```

---

## setParentURL

```
public void setParentURL(java.lang.String parentURL)
```

---

## getType

```
public java.lang.String getType()
```

---

## setType

```
public void setType(java.lang.String type)
```

---

## tud.iir.extraction.qa Class QAUrlStack

```
java.lang.Object
├-- java.util.AbstractCollection
│   ├── java.util.AbstractSet
│   │   ├── java.util.HashSet
│   │   └── tud.iir.extraction.qa.QAUrlStack
```

### All Implemented Interfaces:

java.util.Collection, java.util.Set, java.io.Serializable, java.lang.Cloneable, java.util.Set

---

```
public class QAUrlStack
extends java.util.HashSet
```

---

## Constructors

### QAUrlStack

```
public QAUrlStack()
```

## Methods

### contains

```
public boolean contains(java.lang.Object o)
```

---

Package

**tud.iir.extraction.snippet**

# tud.iir.extraction.snippet

## Class EntitySnippetExtractionThread

```
java.lang.Object
├── java.lang.Thread
│   └── tud.iir.extraction.snippet.EntitySnippetExtractionThread
```

All Implemented Interfaces:  
java.lang.Runnable

```
public class EntitySnippetExtractionThread
extends java.lang.Thread
```

The EntitySnippetExtractionThread extracts snippets for one given entity. Therefore, extracting snippets can be parallelized on the entity level. This class is described in detail in "Friedrich, Christopher. WebSnippets - Extracting and Ranking of entity-centric knowledge from the Web. Diploma thesis, Technische Universität Dresden, April 2010".

**Author:**  
Christopher Friedrich

## Constructors

### EntitySnippetExtractionThread

```
public EntitySnippetExtractionThread(java.lang.ThreadGroup threadGroup,
                                     java.lang.String name,
                                     Entity entity)
```

## Methods

### run

```
public void run()
```



# tud.iir.extraction.snippet

## Class SnippetBuilder

```
java.lang.Object
└--tud.iir.extraction.snippet.SnippetBuilder
```

---

```
public class SnippetBuilder
extends java.lang.Object
```

The SnippetBuilder class provides different snippet extraction techniques through a homogeneous extraction function `extractSnippets()`. Currently implemented are `WEBRESULT_SUMMARY`, `DOCUMENT_SENTENCES` and `DOCUMENT_SNIPPETS`. All these techniques have in common that they receive the Entity and an AggregatedResult as input and return a set of Snippets. This class is described in detail in "Friedrich, Christopher. WebSnippets - Extracting and Ranking of entity-centric knowledge from the Web. Diploma thesis, Technische Universität Dresden, April 2010".

### Author:

Christopher Friedrich

---

## Fields

### WEBRESULT\_SUMMARY

```
public static final int WEBRESULT_SUMMARY
```

Constant value: 0

---

### DOCUMENT\_SENTENCES

```
public static final int DOCUMENT_SENTENCES
```

Constant value: 1

---

### DOCUMENT\_SNIPPETS

```
public static final int DOCUMENT_SNIPPETS
```

Constant value: 2

## Constructors

### SnippetBuilder

```
public SnippetBuilder()
```

## Methods

(continued from last page)

---

## extractSnippets

```
public java.util.List extractSnippets(Entity entity,  
    AggregatedResult webresult,  
    int method)
```

Extract a list of snippets for the provided Entity from the provided AggregatedResult. This function acts as interface to several extraction techniques implemented.

### Parameters:

`entity` - - The entity for which to extract snippets.

`webresult` - - The webresult to extract snippets from.

`method` - - The technique used for extraction. Currently implemented are WEBRESULT\_SUMMARY, DOCUMENT\_SENTENCES and DOCUMENT\_SNIPPETS as described in "Friedrich, Christopher. WebSnippets - Extracting and Ranking of entity-centric knowledge from the Web. Diploma thesis, Technische Universität Dresden, April 2010".

### Returns:

List of snippets

---

## countEntityOccurrences

```
public int countEntityOccurrences(Entity entity,  
    java.lang.String text)
```

Count the occurrences of a certain entity in a provided string.

---

## getEntityChunks

```
public java.util.Set getEntityChunks(Entity entity,  
    java.lang.String text,  
    boolean includePrefixes)
```

Return the set of occurrences of a certain entity in a provided string, including different spellings of the entity. An optional parameter allows to specify whether the entity might be prefixed by "the", "an" or "a".

---

## main

```
public static void main(java.lang.String[] abc)
```

# tud.iir.extraction.snippet

## Class SnippetDuplicateDetection

java.lang.Object

└─tud.iir.extraction.snippet.SnippetDuplicateDetection

```
public class SnippetDuplicateDetection
extends java.lang.Object
```

This class provides different de-duplication techniques to eliminate duplicated or near-duplicated snippets. This class is described in detail in "Friedrich, Christopher. WebSnippets - Extracting and Ranking of entity-centric knowledge from the Web. Diploma thesis, Technische Universität Dresden, April 2010".

**Author:**

Christopher Friedrich

## Fields

### PLAIN

```
public static final int PLAIN
```

Constant value: 0

### SHINGLES

```
public static final int SHINGLES
```

Constant value: 1

## Constructors

### SnippetDuplicateDetection

```
public SnippetDuplicateDetection()
```

## Methods

### removeDuplicates

```
public static void removeDuplicates(java.util.List snippets,
int method)
```

Remove the duplicates from a list of snippets, which are either within the same list or in the database. This might vary by technique. Depending on the technique specified, these are either exact or near duplicates.

**Parameters:**

`snippets` - - List of snippets

`method` - - Technique used to remove duplicates, currently implemented is PLAIN.

# tud.iir.extraction.snippet

## Class SnippetExtractionProcess

```
java.lang.Object
  |
+- java.lang.Thread
    |
    +- tud.iir.extraction.snippet.SnippetExtractionProcess
```

All Implemented Interfaces:  
java.lang.Runnable

---

```
public class SnippetExtractionProcess
extends java.lang.Thread
```

The snippet extraction process.

**Author:**  
Christopher Friedrich

---

## Constructors

### SnippetExtractionProcess

```
public SnippetExtractionProcess()
```

---

### SnippetExtractionProcess

```
public SnippetExtractionProcess(boolean benchmark)
```

---

## Methods

### run

```
public void run()
```

---

### stopExtraction

```
public boolean stopExtraction()
```

---

### isBenchmark

```
public boolean isBenchmark()
```

---

(continued from last page)

## setBenchmark

```
public void setBenchmark(boolean benchmark)
```

# tud.iir.extraction.snippet

## Class SnippetExtractor

```
java.lang.Object
├── tud.iir.extraction.Extractor
│   └── tud.iir.extraction.snippet.SnippetExtractor
```

---

```
public class SnippetExtractor
extends Extractor
```

The SnippetExtractor class extends the Extractor singleton class, retrieves all entities from the knowledge base and schedules k thread runs in parallel, where k is the number of entities. For each entity a separate thread is started. Each thread is a subclass of EntitySnippetExtractionThread. To avoid overloading the system, a threading queue allows to only run i threads in parallel. This class is described in detail in "Friedrich, Christopher. WebSnippets - Extracting and Ranking of entity-centric knowledge from the Web. Diploma thesis, Technische Universität Dresden, April 2010".

### Author:

Christopher Friedrich

---

## Methods

### getInstance

```
public static SnippetExtractor getInstance()
```

---

### startExtraction

```
public void startExtraction()
```

Start extraction of snippets for entities that are fetched from the knowledge base. Continue from last extraction.

---

### startExtraction

```
public void startExtraction(boolean continueFromLastExtraction)
```

---

### main

```
public static void main(java.lang.String[] abc)
```

# tud.iir.extraction.snippet

## Class SnippetFeatureExtractor

java.lang.Object

└─ tud.iir.extraction.snippet.SnippetFeatureExtractor

---

public class **SnippetFeatureExtractor**  
extends java.lang.Object

Given an extracted snippet, a feature vector is generated. This class is described in detail in "Friedrich, Christopher. WebSnippets - Extracting and Ranking of entity-centric knowledge from the Web. Diploma thesis, Technische Universität Dresden, April 2010".

**Author:**

Christopher Friedrich

---

## Constructors

### SnippetFeatureExtractor

public **SnippetFeatureExtractor**()

## Methods

### setFeatures

public static void **setFeatures**([Snippet](#) snippet)

---

### extractPOSFromSentence

public static java.util.List **extractPOSFromSentence**(java.lang.String sentence)

Extract a list of part-of-speech tags from a sentence.

**Parameters:**

sentence - - The sentence

**Returns:**

The part of speech tags.

---

### main

public static void **main**(java.lang.String[] abc)

# tud.iir.extraction.snippet

## Class SnippetQuery

```
java.lang.Object
├── tud.iir.extraction.Query
│   └── tud.iir.extraction.snippet.SnippetQuery
```

```
public class SnippetQuery
    extends Query
```

A snippet query is a search query to retrieve relevant pages for an entity to extract snippets from.

**Author:**

Christopher Friedrich

## Constructors

### SnippetQuery

```
public SnippetQuery(Entity entity)
```

## Methods

### setEntity

```
public void setEntity(Entity entity)
```

### getEntity

```
public Entity getEntity()
```



## tud.iir.extraction.snippet Class SnippetQueryFactory

java.lang.Object

└-tud.iir.extraction.snippet.SnippetQueryFactory

---

public class **SnippetQueryFactory**  
extends java.lang.Object

This class acts as query template builder factory. Given an entity, it generates a set of queries, which are sent to the search engines.

**Author:**

Christopher Friedrich

---

## Methods

### getInstance

public static [SnippetQueryFactory](#) **getInstance**()

---

### createEntityQuery

public [SnippetQuery](#) **createEntityQuery**([Entity](#) entity)

Given an entity, this method returns a SnippetQuery object, which is a set of search engine queries for a given entity.

---

### main

public static void **main**(java.lang.String[] args)

---

Package  
**tud.iir.gui**

# tud.iir.gui

## Class GUIManager

java.lang.Object

└─ tud.iir.gui.GUIManager

All Implemented Interfaces:

java.util.Observer

---

```
public class GUIManager
extends java.lang.Object
implements java.util.Observer
```

The GUIManager manages the complete layout of the WebKnox Core application.

---

## Methods

### getInstance

```
public static GUIManager getInstance()
```

---

### isInstanciaded

```
public static boolean isInstanciaded()
```

---

### update

```
public void update(java.util.Observable o,
    java.lang.Object arg)
```

Get notified when the object changes.

**Parameters:**

- o - The observable object.
- arg - More arguments.

---

### createGUI

```
public void createGUI()
```

---

### main

```
public static void main(java.lang.String[] args)
```

**Parameters:**

(continued from last page)

args

---

## isShowLogging

```
public boolean isShowLogging()
```

---

## setShowLogging

```
public void setShowLogging(boolean showLogging)
```

---

Package  
**tud.iir.helper**

# tud.iir.helper

## Class ArrayHelper

```
java.lang.Object
|
|--tud.iir.helper.ArrayHelper
```

---

```
public class ArrayHelper
extends java.lang.Object
```

Helper functions for common (untyped/generic) arrays.

**Author:**  
Martin Gregor

---

## Constructors

### ArrayHelper

```
public ArrayHelper()
```

## Methods

### removeNullElements

```
public static java.util.ArrayList removeNullElements(java.util.ArrayList array)
```

Removes null objects out of an array.

**Parameters:**  
array

**Returns:**

---

### concat

```
public static java.lang.String[] concat(java.lang.String[] array1,
    java.lang.String[] array2)
```

# tud.iir.helper

## Class CollectionHelper

```
java.lang.Object
└--tud.iir.helper.CollectionHelper
```

---

```
public final class CollectionHelper
extends java.lang.Object
```

This class adds some methods that make it easier to handle collections.

**Author:**  
David Urbansky

---

## Fields

### ASCENDING

```
public static boolean ASCENDING
```

---

### DESCENDING

```
public static boolean DESCENDING
```

---

## Constructors

### CollectionHelper

```
public CollectionHelper()
```

## Methods

### sortByValue

```
public static java.util.LinkedHashMap sortByValue(java.util.Set entrySet)
```

Sort a hashmap by value.

**Parameters:**

`entrySet` - The entry set.

**Returns:**

The sorted map.

---

### sortByValue

```
public static java.util.LinkedHashMap sortByValue(java.util.Set entrySet,
boolean ascending)
```

---

(continued from last page)

Sort a hashmap by value.

**Parameters:**

`entrySet` - The entry set.

`ascending` - Whether to sort ascending or descending.

**Returns:**

The sorted map.

---

## getKeyByValue

```
public static java.lang.Object getKeyByValue(java.util.Map map,  
                                              java.lang.Object value)
```

Get a key given a value (1 to 1 HashMaps)

**Parameters:**

`value` - The value.

**Returns:**

The key that matches the value.

---

## reverse

```
public static java.util.ArrayList reverse(java.util.ArrayList list)
```

---

## getPrint

```
public static java.lang.String getPrint(java.lang.Object[] array)
```

---

## print

```
public static void print(java.lang.Object[] array)
```

---

## print

```
public static void print(java.util.Map map)
```

---

## contains

```
public static boolean contains(java.lang.String[] array,  
                               java.lang.String entry)
```

Check whether a string array contains a string.

**Parameters:**

`array` - The string array.

`entry` - The string entry that is checked against the array.

**Returns:**



(continued from last page)

True, if the entry is contained in the array, false otherwise.

---

## getPrint

```
public static java.lang.String getPrint(java.util.Collection collection)
```

---

## print

```
public static void print(java.util.Collection collection)
```

---

## toHashSet

```
public static java.util.HashSet toHashSet(java.lang.String[] array)
```

# tud.iir.helper

## Class Counter

```
java.lang.Object
└-- tud.iir.helper.Counter
```

---

```
public class Counter
extends java.lang.Object
```

Simple and thread safe up/down counter

**Author:**  
Philipp Katz

---

## Constructors

### Counter

```
public Counter()
```

## Methods

### increment

```
public void increment()
```

---

### decrement

```
public void decrement()
```

---

### increment

```
public void increment(int by)
```

---

### getCount

```
public int getCount()
```

---

### toString

```
public java.lang.String toString()
```

---

## tud.iir.helper Class CountMap

```
java.lang.Object
├-- java.util.AbstractMap
│   └-- java.util.HashMap
│       └-- tud.iir.helper.CountMap
```

### All Implemented Interfaces:

java.util.Map, java.io.Serializable, java.lang.Cloneable, java.util.Map

---

```
public class CountMap
extends java.util.HashMap
```

---

## Constructors

### CountMap

```
public CountMap()
```

---

## Methods

### increment

```
public void increment(java.lang.Object key)
```

---

### get

```
public java.lang.Integer get(java.lang.Object key)
```

## tud.iir.helper Class DataHolder

```
java.lang.Object
|
|--tud.iir.helper.DataHolder
```

---

```
public class DataHolder
extends java.lang.Object
```

The DataHolder can be used to store data objects such as model files. These files do not have to be re-read from hard disk every time they are needed.

**Author:**  
David Urbansky

---

### Constructors

#### DataHolder

```
public DataHolder()
```

### Methods

#### getInstance

```
public static DataHolder getInstance()
```

---

#### containsDataObject

```
public boolean containsDataObject(java.lang.String name)
```

---

#### getDataObject

```
public java.lang.Object getDataObject(java.lang.String name)
```

---

#### putDataObject

```
public void putDataObject(java.lang.String name,
    java.lang.Object object)
```

# tud.iir.helper

## Class DateArrayHelper

java.lang.Object

└─ tud.iir.helper.DateArrayHelper

```
public class DateArrayHelper
extends java.lang.Object
```

Helper functions for arrays consisting extracted dates or subclasses.

**Author:**

Martin Gregor

## Fields

### **FILTER\_IS\_IN\_RANGE**

```
public static final int FILTER_IS_IN_RANGE
```

Filter dates in range (1993 - today).  
Constant value: 0

### **FILTER\_TECH\_URL**

```
public static final int FILTER_TECH_URL
```

Filter URLDates.  
Constant value: 1

### **FILTER\_TECH\_HTTP\_HEADER**

```
public static final int FILTER_TECH_HTTP_HEADER
```

Filter HTTPHeaderDates.  
Constant value: 2

### **FILTER\_TECH\_HTML\_HEAD**

```
public static final int FILTER_TECH_HTML_HEAD
```

Filter HTMLHeadDates.  
Constant value: 3

### **FILTER\_TECH\_HTML\_STRUC**

```
public static final int FILTER_TECH_HTML_STRUC
```

Filter HTMLStructureDates.  
Constant value: 4

### **FILTER\_TECH\_HTML\_CONT**

```
public static final int FILTER_TECH_HTML_CONT
```

(continued from last page)

Filter HTMLContentDates.  
Constant value: 5

---

## FILTER\_TECH\_REFERENCE

```
public static final int FILTER_TECH_REFERENCE
```

Filter ReferenceDates.  
Constant value: 6

---

## FILTER\_TECH\_ARCHIVE

```
public static final int FILTER_TECH_ARCHIVE
```

Filter ArchiveDates.  
Constant value: 7

---

## FILTER\_KEYLOC\_ATTR

```
public static final int FILTER_KEYLOC_ATTR
```

Filter contentDates with key-location in attribute.  
Constant value: 201

---

## FILTER\_KEYLOC\_CONT

```
public static final int FILTER_KEYLOC_CONT
```

Filter contentDates with key-location in content.  
Constant value: 202

---

## FILTER\_KEYLOC\_NO

```
public static final int FILTER_KEYLOC_NO
```

Filter contentDates without key in attribute nor content.  
Constant value: 203

---

## FILTER\_FULL\_DATE

```
public static final int FILTER_FULL_DATE
```

Filter dates with year, month and day.  
Constant value: 204

## Constructors

### DateArrayHelper

```
public DateArrayHelper()
```

## Methods

### filter

```
public static java.util.ArrayList filter(java.util.ArrayList dates,  
int filter)
```

(continued from last page)

Filters an array-list.

**Parameters:**

dates  
filter

**Returns:**

---

**filter**

```
public static java.util.HashMap filter(java.util.HashMap dates,  
int filter)
```

---

**filterFormat**

```
public static java.util.ArrayList filterFormat(java.util.ArrayList dates,  
java.lang.String format)
```

---

**arrangeByDate**

```
public static java.util.ArrayList arrangeByDate(java.util.ArrayList dates,  
int stopFlag)
```

Group equal dates in array lists.

E.g. d1=May 2010; d2=05.2010; d3=01.05.10; d4=01st May '10 --> (d1&d2) & (d3&d4).

Every date can be only in one group.

A group is a array list of dates.

**Parameters:**

dates - Arraylist of dates.

**Returns:**

A arraylist of groups, that are arraylists too.

---

**arrangeByDate**

```
public static java.util.ArrayList arrangeByDate(java.util.ArrayList dates)
```

Group equal dates in array lists.

E.g. d1=May 2010; d2=05.2010; d3=01.05.10; d4=01st May '10 --> (d1&d2) & (d3&d4).

Every date can be only in one group.

A group is a array list of dates.

**Parameters:**

dates - Arraylist of dates.

**Returns:**

A arraylist of groups, that are arraylists too.

---

**arrangeMapByDate**

```
public static java.util.ArrayList arrangeMapByDate(java.util.HashMap dates)
```

Orders a map by dates.

(continued from last page)

**Parameters:**

dates

**Returns:**

---

## arrangeMapByDate

```
public static java.util.ArrayList arrangeMapByDate(java.util.HashMap dates,
    int stopFlag)
```

---

## countDates

```
public static int countDates(java.lang.Object date,
    java.util.ArrayList dates)
```

---

## countDates

```
public static int countDates(java.lang.Object date,
    java.util.HashMap dates)
```

Count equal dates.

**Parameters:**

date

dates

**Returns:**

---

## countDates

```
public static int countDates(java.lang.Object date,
    java.util.HashMap dates,
    int stopFlag)
```

---

## printDateArray

```
public static void printDateArray(java.util.ArrayList dates,
    int filterTechnique,
    java.lang.String format)
```

Same as printDateArray() with filter of techniques. These are found in ExtracedDate as static properties.  
And a format, found as second value of RegExp.

**Parameters:**

dates

filterTechnique

format



(continued from last page)

## printDateArray

```
public static void printDateArray(java.util.ArrayList dates)
```

System.out.println for each date in dates, with some properties.

### Parameters:

dates

---

## printDateArray

```
public static void printDateArray(java.util.ArrayList dates,  
    int filterTechnique)
```

Same as printDateArray() with filter of techniques. These are found in ExtracedDate as static properties.

### Parameters:

dates

filterTechnique

---

## removeFormat

```
public static java.util.ArrayList removeFormat(java.util.ArrayList dates,  
    java.lang.String format)
```

Remove dates from the array.

### Parameters:

dates

format

### Returns:

---

## printDateMap

```
public static void printDateMap(java.util.Map.Entry[] dateMap)
```

---

## printDateMap

```
public static void printDateMap(java.util.Map.Entry[] dateMap,  
    int filter)
```

---

## printDateMap

```
public static void printDateMap(java.util.HashMap dateMap)
```

---

## printDateMap

```
public static void printDateMap(java.util.HashMap dateMap,  
    int filter)
```

---

(continued from last page)

---

## getRatedDates

```
public static java.util.ArrayList getRatedDates(java.util.HashMap dates,  
double rate)
```

Returns an array of dates, that have a given rate.

### Parameters:

dates  
rate

### Returns:

---

## getRatedDates

```
public static java.util.ArrayList getRatedDates(java.util.HashMap dates,  
double rate,  
boolean include)
```

---

## getSameDates

```
public static java.util.ArrayList getSameDates(ExtractedDate date,  
java.util.ArrayList dates)
```

Returns an array of dates that are equal to a given date.

### Parameters:

date  
dates

### Returns:

---

## getSameDates

```
public static java.util.ArrayList getSameDates(ExtractedDate date,  
java.util.ArrayList dates,  
int stopFlag)
```

---

## getSameDatesMap

```
public static java.util.HashMap getSameDatesMap(ExtractedDate date,  
java.util.HashMap dates)
```

Returns a hashmap of date are equal to given date.

### Parameters:

date  
dates

### Returns:

---

## getSameDatesMap

```
public static java.util.HashMap getSameDatesMap(ExtractedDate date,  
        java.util.HashMap dates,  
        int stopFlag)
```

---

## getDifferentDatesMap

```
public static java.util.HashMap getDifferentDatesMap(ExtractedDate date,  
        java.util.HashMap dates)
```

---

## getDifferentDatesMap

```
public static java.util.HashMap getDifferentDatesMap(ExtractedDate date,  
        java.util.HashMap dates,  
        int stopFlag)
```

---

## orderHashMap

```
public static java.util.Map.Entry[] orderHashMap(java.util.HashMap dates)
```

Order by rate.

### Parameters:

dates

### Returns:

---

## orderHashMap

```
public static java.util.Map.Entry[] orderHashMap(java.util.HashMap dates,  
        boolean reverse)
```

Order by rate. Lowest is first.

### Parameters:

dates

reverse

### Returns:

---

## isAllZero

```
public static boolean isAllZero(java.util.HashMap dates)
```

---

## getExactestDates

```
public static java.util.ArrayList getExactestDates(java.util.HashMap dates)
```

---

## getExactestMap

```
public static java.util.HashMap getExactestMap(java.util.HashMap dates)
```

---

## getHighestRate

```
public static double getHighestRate(java.util.HashMap dates)
```

Returns the highest rate in a map.

**Parameters:**

dates

**Returns:**

---

## getFirstElement

```
public static java.lang.Object getFirstElement(java.util.HashMap map)
```

Returns first element of a hashmap.

**Parameters:**

map

**Returns:**

# tud.iir.helper

## Class DateComparator

java.lang.Object

└─ tud.iir.helper.DateComparator

### All Implemented Interfaces:

java.util.Comparator

---

public class **DateComparator**  
extends java.lang.Object  
implements java.util.Comparator

---

## Fields

### STOP\_YEAR

public static final int **STOP\_YEAR**

Compare will stop after year. Value = 1.  
Constant value: 1

---

### STOP\_MONTH

public static final int **STOP\_MONTH**

Compare will stop after month. Value = 2.  
Constant value: 2

---

### STOP\_DAY

public static final int **STOP\_DAY**

Compare will stop after day. Value = 3.  
Constant value: 3

---

### STOP\_HOUR

public static final int **STOP\_HOUR**

Compare will stop after hour. Value = 4.  
Constant value: 4

---

### STOP\_MINUTE

public static final int **STOP\_MINUTE**

Compare will stop after minute. Value = 5.  
Constant value: 5

---

(continued from last page)

## STOP\_SECOND

```
public static final int STOP_SECOND
```

Compare will not stop. (After second there are no more comparable values. Value = 6.  
Constant value: 6

## MEASURE\_MILLI\_SEC

```
public static final int MEASURE_MILLI_SEC
```

Get date-difference in milliseconds  
Constant value: 1

## MEASURE\_SEC

```
public static final int MEASURE_SEC
```

Get date-difference in seconds  
Constant value: 1000

## MEASURE\_MIN

```
public static final int MEASURE_MIN
```

Get date-difference in minutes  
Constant value: 60000

## MEASURE\_HOUR

```
public static final int MEASURE_HOUR
```

Get date-difference in hours  
Constant value: 3600000

## MEASURE\_DAY

```
public static final int MEASURE_DAY
```

Get date-difference in days  
Constant value: 86400000

## Constructors

### DateComparator

```
public DateComparator()
```

## Methods

### compare

```
public int compare(ExtractedDate date1,  
                  ExtractedDate date2)
```

(continued from last page)

Compares two dates.

Returns -1, 0 or 1 if date1 is newer, equals or older then date2.

If both dates are not comparable, for e.g. date1.month is not set, the returning value will be -2.

This does only matter, if the higher parameter are equal.

For e.g.:

date1.year = 2007 and date2.year = 2006; date1.month=11 and date2.month =-1.

Then the returning value will be -1, because 2007>2006.

If date1.year is 2006 as well, then the return value will be -2, because the years are equal and the month can not be compared.

---

## compare

```
public int compare(ExtractedDate date1,  
                  ExtractedDate date2,  
                  int stopFlag)
```

Like **compare([ExtractedDate](#) date1, [ExtractedDate](#) date2)**, but compares only until a given depth.

For e.g. usually 12.04.2007 and April 2007 can not be compared. But with stopflag STOP\_DAY only year and month will be compared.

So normal compare would return -2, but this time the result is 0.

### Parameters:

date1

date2

stopFlag - Depth of comparing. Values are given as static constant in this class. (STOP\_...)

### Returns:

---

## compare

```
public int compare(ExtractedDate date1,  
                  ExtractedDate date2,  
                  boolean ignoreComparable)
```

---

## compare

```
public int compare(ExtractedDate date1,  
                  ExtractedDate date2,  
                  boolean ignoreComparable,  
                  int compareDepth)
```

---

## compare

```
public int compare(int i,  
                  int k)
```

Compares a parameter of two dates. (date1.getYear() and date2.getYear()).

If i or k equals -1, then -2 will be returned.

Otherwise -1 for i > k, 0 for i=k, 1 for i < k;

If k=i=-1 -> 0 will be returned.

### Parameters:

i

k

### Returns:

## getCompareDepth

```
public int getCompareDepth(ExtractedDate date1,  
    ExtractedDate date2)
```

Finds out, until which depth two dates are comparable.  
Order is year, month, day, hour, minute and second.

**Parameters:**

date1  
date2

**Returns:**

Integer with the value of stop\_property. Look for it in static properties.

---

## getDifference

```
public double getDifference(ExtractedDate date1,  
    ExtractedDate date2,  
    int measure)
```

Returns the difference between two extracted dates.  
If dates can not be compared -1 will be returned.  
Otherwise difference is calculated to maximal possible depth. (year-month-day-hour-minute-second).  
Measures of returning value can be set to milliseconds, seconds, minutes, hours and days. There for use static properties.

**Parameters:**

date1  
date2  
measure - Found in DateComparator.

**Returns:**

A positive (absolute) difference. To know which date is more actual use **compare**.

---

## getEqualDate

```
public java.util.ArrayList getEqualDate(java.lang.Object date,  
    java.util.ArrayList dates)
```

Filters a set of dates out of an array, that have same extraction date like a given date.

**Parameters:**

date - defines the extraction date.  
dates - array to be filtered.

**Returns:**

Array of dates, that are equal to the date.

---

## orderDates

```
public java.util.ArrayList orderDates(java.util.ArrayList dates)
```

---



(continued from last page)

## orderDates

```
public java.util.ArrayList orderDates(java.util.ArrayList dates,  
    boolean reverse)
```

---

## orderDates

```
public java.util.ArrayList orderDates(java.util.HashMap dates)
```

---

## orderDates

```
public java.util.ArrayList orderDates(java.util.HashMap dates,  
    boolean reverse)
```

---

## orderDatesArray

```
public java.lang.Object[] orderDatesArray(java.util.ArrayList dates)
```

Orders a datelist, beginning with oldest date.

**Parameters:**

dates

**Returns:**

---

## getOldestDate

```
public java.lang.Object getOldestDate(java.util.HashMap dates)
```

---

## getYoungestDate

```
public java.lang.Object getYoungestDate(java.util.HashMap dates)
```

---

## getOldestDate

```
public java.lang.Object getOldestDate(java.util.ArrayList dates)
```

---

## getYoungestDate

```
public java.lang.Object getYoungestDate(java.util.ArrayList dates)
```

# tud.iir.helper

## Class DateHelper

java.lang.Object

└─ tud.iir.helper.DateHelper

---

```
public class DateHelper
extends java.lang.Object
```

This class helps to transform and help with dates.

**Author:**

David Urbansky

---

## Fields

### SECOND\_MS

```
public static final int SECOND_MS
```

Constant value: 1000

---

### MINUTE\_MS

```
public static final int MINUTE_MS
```

Constant value: 60000

---

### HOURL\_MS

```
public static final int HOURL_MS
```

Constant value: 3600000

---

### DAY\_MS

```
public static final int DAY_MS
```

Constant value: 86400000

---

### WEEK\_MS

```
public static final int WEEK_MS
```

Constant value: 604800000

---

### MONTH\_MS

```
public static final int MONTH_MS
```

---

(continued from last page)

Constant value: **-1702967296**

---

## YEAR\_MS

```
public static final int YEAR_MS
```

Constant value: **1471228928**

## Constructors

### DateHelper

```
public DateHelper()
```

## Methods

### containsDate

```
public static boolean containsDate(java.lang.String searchString)
```

---

### getCurrentDatetime

```
public static java.lang.String getCurrentDatetime(java.lang.String format)
```

---

### getDatetime

```
public static java.lang.String getDatetime(java.lang.String format,  
                                           long timestamp)
```

---

### getTimeOfDay

```
public static long getTimeOfDay(java.util.Date date,  
                                int resolution)
```

Get the number of hours, minutes, seconds, or milliseconds that passed on the given day from midnight.

**Parameters:**

`date` - The date of the day including time.

`resolution` - The resolution (Calendar.HOUR, Calendar.MINUTE, Calendar.SECOND or Calendar.MILLISECOND)

**Returns:**

A positive number of the passed time.

---

### getTimeOfDay

```
public static long getTimeOfDay(long timestamp,  
                                int resolution)
```

---

## getCurrentDatetime

```
public static java.lang.String getCurrentDatetime()
```

Return the current date as a string with the format "yyyy-MM-dd\_HH-mm-ss".

**Returns:**

The date as a string.

---

## monthNameToNumber

```
public static java.lang.String monthNameToNumber(java.lang.String monthName)
```

---

## getRuntime

```
public static java.lang.String getRuntime(long startTime)
```

Returns the time that passed since the start time.

**Parameters:**

`startTime` - A timestamp.

**Returns:**

The passed time since the time of the timestamp. The format is Hh:Mm:Ss:YYms.

---

## getRuntime

```
public static java.lang.String getRuntime(long startTime,  
long stopTime)
```

---

## getRuntime

```
public static java.lang.String getRuntime(long startTime,  
long stopTime,  
boolean output)
```

---

## getTimeString

```
public static java.lang.String getTimeString(long time)
```

---

## getTimestamp

```
public static long getTimestamp(java.lang.String date)
```

Create the UNIX timestamp for the given date (UTC).

**Parameters:**

`normalizedDate` - A date in normalized form: yyyy-MM-dd [hh:mm:ss[.f]]

---

(continued from last page)

**Returns:**

The UNIX timestamp for that date.

---

**main**

```
public static void main(java.lang.String[] t)
```

# tud.iir.helper

## Class DBStore

```
java.lang.Object
└-- tud.iir.helper.DBStore
```

---

```
public class DBStore
extends java.lang.Object
```

This class allows one to save data into a database instead of keeping it in memory.

**Author:**  
David Urbansky

---

## Constructors

### DBStore

```
public DBStore(java.lang.String tableName)
```

---

### DBStore

```
public DBStore(java.lang.String tableName,
               java.lang.String dbUsername,
               java.lang.String dbPassword)
```

## Methods

### clear

```
public void clear()
```

Empty the dbstore.

---

### get

```
public java.lang.Object get(java.lang.String key)
```

Read the word from the unnormalized table with all information (faster).

**Parameters:**

`word` - The word to look up.

**Returns:**

The category entries for the word.

---

### getByKey

```
public java.lang.Object getByKey(java.lang.String key)
```

---

## put

```
public void put(java.lang.String key,  
               int value)
```

---

## put

```
public void put(java.lang.String key,  
               double value)
```

---

## put

```
public void put(java.lang.String key,  
               java.lang.String value)
```

---

## remove

```
public void remove(java.lang.String key)
```

---

## getDbType

```
public java.lang.String getDbType()
```

---

## setDbType

```
public void setDbType(java.lang.String dbType)
```

---

## getDbDriver

```
public java.lang.String getDbDriver()
```

---

## setDbDriver

```
public void setDbDriver(java.lang.String dbDriver)
```

---

## getDbHost

```
public java.lang.String getDbHost()
```

---

---

## setDbHost

```
public void setDbHost(java.lang.String dbHost)
```

---

---

## getDbPort

```
public java.lang.String getDbPort()
```

---

---

## setDbPort

```
public void setDbPort(java.lang.String dbPort)
```

---

---

## getTableName

```
public java.lang.String getTableName()
```

---

---

## setTableName

```
public void setTableName(java.lang.String tableName)
```

---

---

## getDbUsername

```
public java.lang.String getDbUsername()
```

---

---

## setDbUsername

```
public void setDbUsername(java.lang.String dbUsername)
```

---

---

## getDbPassword

```
public java.lang.String getDbPassword()
```

---

---

## setDbPassword

```
public void setDbPassword(java.lang.String dbPassword)
```

---

---

## size

```
public int size()
```

---



## main

```
public static void main(java.lang.String[] args)
```

### Parameters:

args

# tud.iir.helper

## Class FileHelper

```
java.lang.Object
└-- tud.iir.helper.FileHelper
```

---

```
public class FileHelper
extends java.lang.Object
```

The FileHelper helps with file concerning tasks.

**Author:**

David Urbansky, Philipp Katz, Martin Werner

---

## Constructors

### FileHelper

```
public FileHelper()
```

## Methods

### isFileName

```
public static boolean isFileName(java.lang.String name)
```

Checks if is file name.

**Parameters:**

name - the name

**Returns:**

true, if is file name

---

### isVideoFile

```
public static boolean isVideoFile(java.lang.String fileType)
```

Checks if is video file.

**Parameters:**

fileType - the file type

**Returns:**

true, if is video file

---

### isAudioFile

```
public static boolean isAudioFile(java.lang.String fileType)
```

Checks if is audio file.

---

(continued from last page)

**Parameters:**`fileType` - the file type**Returns:**`true`, if is audio file

---

## getFilePath

```
public static java.lang.String getFilePath(java.lang.String path)
```

Gets the file path.

**Parameters:**`path` - the path**Returns:**the file path

---

## getFileName

```
public static java.lang.String getFileName(java.lang.String path)
```

Gets the file name.

**Parameters:**`path` - the path**Returns:**the file name

---

## appendToFileName

```
public static java.lang.String appendToFileName(java.lang.String filePath,  
java.lang.String appendix)
```

## getFileType

```
public static java.lang.String getFileType(java.lang.String path)
```

Gets the file type.

**Parameters:**`path` - the path**Returns:**the file type

---

## readHTMLFileToString

```
public static java.lang.String readHTMLFileToString(java.lang.String path,  
boolean stripTags)
```

Read html file to string.

**Parameters:**`path` - the path`stripTags` - the strip tags

---

(continued from last page)

**Returns:**  
the string

---

## readFileToString

```
public static java.lang.String readFileToString(java.lang.String path)
```

Read file to string.

**Parameters:**  
path - the path

**Returns:**  
the string

---

## readFileToArray

```
public static java.util.List readFileToArray(java.lang.String path)
```

Create a list with each line of the given file as an element.

**Parameters:**  
path - The path of the file.

**Returns:**  
A list with the lines as elements.

---

## readFileToArray

```
public static java.util.List readFileToArray(java.net.URL fileURL)
```

.

**Parameters:**  
fileURL - the file url

**Returns:**  
the list

---

## readFileToArray

```
public static java.util.List readFileToArray(java.io.File contentFile)
```

Create a list with each line of the given file as an element.

**Parameters:**  
contentFile - the content file

**Returns:**  
A list with the lines as elements.

---

(continued from last page)

---

## fileContentToLines

```
public static void fileContentToLines(java.lang.String inputFilePath,  
    java.lang.String outputFilePath,  
    java.lang.String separator)
```

Split the contents of a file into lines. For example: a, b, c becomes a b c when the separator is ",".

### Parameters:

inputFilePath - The input file.  
outputFilePath - Where the transformed file should be saved.  
separator - The separator that is used to split.

---

## removeDuplicateLines

```
public static void removeDuplicateLines(java.lang.String inputFilePath,  
    java.lang.String outputFilePath)
```

Remove identical lines for the given input file and save it to the output file.

### Parameters:

inputFilePath - The input file.  
outputFilePath - Where the transformed file should be saved.

---

## performActionOnEveryLine

```
public static int performActionOnEveryLine(java.lang.String filePath,  
    LineAction la)
```

Perform action on every line.

### Parameters:

filePath - the file path  
la - the la

### Returns:

the int

---

## writeToFile

```
public static void writeToFile(java.lang.String filePath,  
    java.lang.StringBuilder string)
```

Write to file.

### Parameters:

filePath - the file path  
string - the string

---

## writeToFile

```
public static void writeToFile(java.lang.String filePath,  
    java.util.Collection lines)
```

Writes a Collection of Objects to a file. Each Object's {`Object.toString()`} invocation represents a line.

### Parameters:

filePath - the file path  
lines - the lines

## writeToFile

```
public static void writeToFile(java.lang.String filePath,  
                               java.lang.String string)
```

Write to file.

### Parameters:

filePath - the file path  
string - the string

---

## appendToFile

```
public static void appendToFile(java.lang.String filePath,  
                                 java.lang.StringBuilder string,  
                                 boolean before)
```

Add some text to a file. TODO Attention -- when appending to files too big for memory this method will cause data loss. readFileToString will read until memory runs out (catching OutOfMemoryError) and return the partially read content, appendToFile then appends to the partial content and writes it back to disk. I have added the two methods appendFile/prependFile which use buffers instead of reading the whole files in memory. -- Philipp, 2010-07-10.

### Parameters:

filePath - The path to the file.  
string - The text to append.  
before - If true, the text will be appended before all other content, if false it will be appended to the end of the file.

---

## appendToFile

```
public static void appendToFile(java.lang.String filePath,  
                                 java.lang.String string,  
                                 boolean before)
```

Append to file.

### Parameters:

filePath - the file path  
string - the string  
before - the before

---

## appendFile

```
public static void appendFile(java.lang.String filePath,  
                               java.lang.String stringToAppend)  
throws java.io.IOException
```

Appends (i. e. inserts at the end) a String to the specified File.

### Parameters:

filePath - the file path  
stringToAppend - the string to append

### Throws:

IOException - Signals that an I/O exception has occurred.

---

(continued from last page)

## prependFile

```
public static void prependFile(java.lang.String filePath,  
    java.lang.String stringToPrepend)  
    throws java.io.IOException
```

Prepends (i. e. inserts at the beginning) a String to the specified File. Inspired by <http://stackoverflow.com/questions/2537944/prepend-lines-to-file-in-java>

### Parameters:

filePath - the file path  
stringToPrepend - the string to prepend

### Throws:

IOException - Signals that an I/O exception has occurred.

---

## deserialize

```
public static java.lang.Object deserialize(java.lang.String filePath)
```

Deserialize.

### Parameters:

filePath - the file path

### Returns:

the object

---

## serialize

```
public static void serialize(java.io.Serializable obj,  
    java.lang.String filePath)
```

Serialize.

### Parameters:

obj - the obj  
filePath - the file path

---

## rename

```
public static java.lang.String rename(java.io.File inputFile,  
    java.lang.String newName)
```

Rename.

### Parameters:

inputFile - the input file  
newName - the new name

### Returns:

the string

---

## copyFile

```
public static void copyFile(java.lang.String sourceFile,  
    java.lang.String destinationFile)
```

Copy a file.

---

(continued from last page)

**Parameters:**

sourceFile - The file to copy.  
destinationFile - The destination of the file.

---

## copyDirectory

```
public static void copyDirectory(java.lang.String srcPath,  
    java.lang.String dstPath)
```

Copy directory.

**Parameters:**

srcPath - the src path  
dstPath - the dst path

---

## copyDirectory

```
public static void copyDirectory(java.io.File srcPath,  
    java.io.File dstPath)
```

Copy directory.

**Parameters:**

srcPath - the src path  
dstPath - the dst path

---

## delete

```
public static boolean delete(java.lang.String filename,  
    boolean deleteNonEmptyDirectory)
```

Delete a file or a directory.

**Parameters:**

filename - The name of the file or directory.  
deleteNonEmptyDirectory - If true, and filename is a directory, it will be deleted with all its contents.

**Returns:**

True if the deletion was successful, false otherwise.

---

## delete

```
public static boolean delete(java.lang.String filename)
```

Delete.

**Parameters:**

filename - the filename

**Returns:**

true, if successful

---

## cleanDirectory

```
public static boolean cleanDirectory(java.lang.String dirPath)
```

Delete all files inside a directory.



(continued from last page)

**Parameters:**`dirPath` - the directoryPath**Returns:**true, if successful

---

## move

```
public static boolean move(java.io.File file,  
    java.lang.String newPath)
```

Move.

**Parameters:**`file` - the file`newPath` - the new path**Returns:**true, if successful

---

## addFileHeader

```
public static void addFileHeader(java.lang.String folderPath,  
    java.lang.StringBuilder header)
```

Add a header to all files from a certain folder.

**Parameters:**`folderPath` - The path to the folder.`header` - The header text to append.

## getFiles

```
public static java.io.File[] getFiles(java.lang.String folderPath)
```

Get all files from a certain folder.

**Parameters:**`folderPath` - The path to the folder.**Returns:**An array of files that are in that folder.

---

## getFiles

```
public static java.io.File[] getFiles(java.lang.String folderPath,  
    java.lang.String substring)
```

Gets the files.

**Parameters:**`folderPath` - the folder path`substring` - the substring**Returns:**the files

---

(continued from last page)

---

## getNumberOfLines

```
public static int getNumberOfLines(java.lang.String fileName)
```

Get the number of lines in an ASCII document.

### Parameters:

fileName - The name of the file.

### Returns:

The number of lines.

---

## zip

```
public static boolean zip(java.lang.String text,  
    java.lang.String filenameOutput)
```

Zip some text and save to a file.

[http://www.java2s.com/Tutorial/Java/0180\\_\\_File/ZipafilewithGZIPOutputStream.htm](http://www.java2s.com/Tutorial/Java/0180__File/ZipafilewithGZIPOutputStream.htm)

### Parameters:

text - The text to be zipped.

filenameOutput - The name of the zipped file.

### Returns:

True if zipping and saving was successfully, false otherwise.

---

## zipString

```
public static java.lang.String zipString(java.lang.String text)
```

Zip string.

### Parameters:

text - the text

### Returns:

the string

---

## unzipFile

```
public static void unzipFile(java.lang.String filenameInput,  
    java.lang.String filenameOutput)
```

Unzip a file.

### Parameters:

filenameInput - The name of the zipped file.

filenameOutput - The target name of the unzipped file.

---

## unzipFile

```
public static void unzipFile(java.lang.String filenameInput)
```

Unzip file.

### Parameters:

filenameInput - the filename input

---

## unzipFile7z

```
public static void unzipFile7z(java.lang.String filenameInput)
```

Unzip file7z.

**Parameters:**

filenameInput - the filename input

---

## unzipFileCmd

```
public static void unzipFileCmd(java.lang.String filenameInput,  
    java.lang.String consoleCommand)
```

Unzip file cmd.

**Parameters:**

filenameInput - the filename input

consoleCommand - the console command

---

## unzipFileToString

```
public static java.lang.String unzipFileToString(java.lang.String filename)
```

Unzip a file and return the unzipped string.

**Parameters:**

filename - The name of the zipped file.

**Returns:**

The unzipped content of the file.

---

## unzipInputStreamToString

```
public static java.lang.String unzipInputStreamToString(java.io.InputStream in)
```

Unzip a input stream to string.

**Parameters:**

in - The input stream with the zipped content.

**Returns:**

The unzipped string.

---

## fileExists

```
public static boolean fileExists(java.lang.String filePath)
```

File exists.

**Parameters:**

filePath - the file path

**Returns:**

true, if successful

---

(continued from last page)

## createDirectory

```
public static boolean createDirectory(java.lang.String directoryPath)
```

---

## main

```
public static void main(java.lang.String[] a)
```

The main method.

### Parameters:

a - the arguments

# tud.iir.helper

## Class HTMLHelper

```
java.lang.Object
|
+--tud.iir.helper.HTMLHelper
```

---

```
public class HTMLHelper
extends java.lang.Object
```

Some HTML specific helper methods.

**Author:**

David Urbansky, Martin Werner, Philipp Katz, Martin Gregor

---

## Methods

### countTags

```
public static int countTags(java.lang.String htmlText)
```

Count the tags.

**Parameters:**

htmlText - The html text.

**Returns:**

The number of tags.

---

### countTagLength

```
public static int countTagLength(java.lang.String taggedText)
```

Count the number of characters used for tags in the given string.

For example, <PHONE>iphone 4</PHONE> ==> 15

**Parameters:**

taggedText - The text with tags.

**Returns:**

The cumulated number of characters used for tags in the given text.

---

### countTags

```
public static int countTags(java.lang.String htmlText,
                             boolean distinct)
```

Count tags.

**Parameters:**

htmlText - The html text.

distinct - If true, count multiple occurrences of the same tag only once.

---

(continued from last page)

**Returns:**

The number of tags.

---

## removeHTMLTags

```
public static java.lang.String removeHTMLTags(java.lang.String htmlContent,  
        boolean stripTags,  
        boolean stripComments,  
        boolean stripJSAndCSS,  
        boolean joinTagsAndRemoveNewlines)
```

Remove all style and script tags including their content (css, javascript). Remove all other tags as well. Close gaps.

**Parameters:**

htmlContent - the html content  
stripTags - the strip tags  
stripComments - the strip comments  
stripJSAndCSS - the strip js and css  
joinTagsAndRemoveNewlines - the join tags and remove newlines

**Returns:**

The text of the web page.

---

## removeHTMLTags

```
public static java.lang.String removeHTMLTags(java.lang.String htmlContent)
```

---

## removeConcreteHTMLTag

```
public static java.lang.String removeConcreteHTMLTag(java.lang.String pageString,  
        java.lang.String tag)
```

Removes the concrete html tag.

**Parameters:**

pageContent - The html text.  
tag - The tag that should be removed.

**Returns:**

The html text without the tag.

---

## removeConcreteHTMLTag

```
public static java.lang.String removeConcreteHTMLTag(java.lang.String pageContent,  
        java.lang.String beginTag,  
        java.lang.String endTag)
```

Remove concrete HTMLTags from a string; this version is for special-tags like .

**Parameters:**

pageContent - The html text.  
beginTag - The begin tag.  
endTag - The end tag.

**Returns:**

The string without the specified html tag.

## getConcreteTags

```
public static java.util.List getConcreteTags(java.lang.String pageString,  
                                             java.lang.String tag)
```

Get a list of concrete HTMLTags; begin- and endtag are not different.

**Parameters:**

pageContent - The html text.  
tag - The tag.

**Returns:**

A list of concrete tags.

---

## getConcreteTags

```
public static java.util.List getConcreteTags(java.lang.String pageString,  
                                             java.lang.String beginTag,  
                                             java.lang.String endTag)
```

Get a list of concrete HTMLTags; its possible that begin- and endtag are different like .

**Parameters:**

pageString - The html text.  
beginTag - The begin tag.  
endTag - The end tag.

**Returns:**

A list of concrete tag names.

---

## htmlToString

```
public static java.lang.String htmlToString(org.w3c.dom.Node node)
```

Converts HTML markup to a more or less human readable string. For example we insert line breaks for HTML block level elements, filter out comments, scripts and stylesheets, remove unnecessary white space and so on. In contrast to [@link removeHTMLTags\(String, boolean, boolean, boolean, boolean\)](#), which works on Strings and just strips out all tags via RegExes, this approach tries to keep some structure for displaying HTML content in text mode in a readable form.

**Parameters:**

node

**Returns:**

---

## htmlToString

```
public static java.lang.String htmlToString(java.lang.String html,  
                                             boolean oneLine)
```

Allows to strip HTML tags from HTML fragments. It will use the Neko parser to parse the String first and then remove the tags, based on the document's structure. Advantage instead of using RegExes to strip the tags is, that whitespace is handled more correctly than in [removeHTMLTags\(String, boolean, boolean, boolean, boolean\)](#) which never worked well for me.

**Parameters:**

html  
oneLine

---

(continued from last page)

**Returns:**

---

## extractTagElement

```
public static java.lang.String extractTagElement(java.lang.String pattern,  
        java.lang.String content,  
        java.lang.String removeTerm)
```

Extract values e.g for: src=, href= or title=

**Parameters:**

pattern - the pattern

content - the content

removeTerm - the term which should be removed e.g. " or '

**Returns:**

the string

---

## isSimpleElement

```
public static boolean isSimpleElement(org.w3c.dom.Node node)
```

Checks, if a node is simple like &lt;u>,&lt;b>,&lt;i>,...

**Parameters:**

node

**Returns:**

true if simple, else false.

---

## isHeadlineTag

```
public static boolean isHeadlineTag(java.lang.String tag)
```

Checks, if tag is a headline.

**Parameters:**

tag

**Returns:**

---

## isHeadlineTag

```
public static boolean isHeadlineTag(org.w3c.dom.Node tag)
```

---

## replaceHTMLSymbols

```
public static java.lang.String replaceHTMLSymbols(java.lang.String text)
```

---



(continued from last page)

## main

```
public static void main(java.lang.String[] args)
    throws java.lang.Exception
```

# tud.iir.helper

## Class LineAction

java.lang.Object

└─ tud.iir.helper.LineAction

public abstract class **LineAction**  
extends java.lang.Object

## Fields

### arguments

public java.lang.Object **arguments**

## Constructors

### LineAction

public **LineAction**()

### LineAction

public **LineAction**(java.lang.Object[] parameters)

## Methods

### performAction

public abstract void **performAction**(java.lang.String line,  
int lineNumber)

### breakLineLoop

public void **breakLineLoop**()

## tud.iir.helper

# Class ListSimilarity

java.lang.Object

└--tud.iir.helper.ListSimilarity

---

public class **ListSimilarity**  
extends java.lang.Object

---

## Constructors

### ListSimilarity

public **ListSimilarity**()

## Methods

### setShiftSimilartiy

public void **setShiftSimilartiy**(double shiftSimilartiy)

---

### getShiftSimilartiy

public double **getShiftSimilartiy**()

---

### setSquaredShiftSimilartiy

public void **setSquaredShiftSimilartiy**(double squaredShiftSimilartiy)

---

### getSquaredShiftSimilartiy

public double **getSquaredShiftSimilartiy**()

---

### getRmse

public double **getRmse**()

---

(continued from last page)

## setRmse

```
public void setRmse(double rmse)
```

## tud.iir.helper

# Class LoggerMessage

```
java.lang.Object
└--tud.iir.helper.LoggerMessage
```

---

```
public class LoggerMessage
extends java.lang.Object
```

The LoggerMessage is a message that is sent from the Logger to its observers.

**Author:**

David Urbansky

---

## Constructors

### LoggerMessage

```
public LoggerMessage()
```

## Methods

### getLoggerName

```
public java.lang.String getLoggerName()
```

### setLoggerName

```
public void setLoggerName(java.lang.String loggerName)
```

### getMessage

```
public java.lang.String getMessage()
```

### getMessage

```
public java.lang.String getMessage(boolean addBreak)
```

### setMessage

```
public void setMessage(java.lang.String message)
```

# tud.iir.helper

## Class MathHelper

```
java.lang.Object
|
|--tud.iir.helper.MathHelper
```

---

```
public class MathHelper
extends java.lang.Object
```

The MathHelper adds mathematical functionality.

**Author:**  
David Urbansky

---

## Constructors

### MathHelper

```
public MathHelper()
```

## Methods

### round

```
public static double round(double number,
                           int digits)
```

---

### getPower

```
public static int getPower(java.lang.String numberString)
```

---

### isWithinMargin

```
public static boolean isWithinMargin(double value1,
                                       double value2,
                                       double margin)
```

---

### isWithinCorrectnessMargin

```
public static boolean isWithinCorrectnessMargin(double questionedValue,
                                                  double correctValue,
                                                  double correctnessMargin)
```

---

(continued from last page)

---

## faculty

```
public static int faculty(int number)
```

---

## getMedianDifference

```
public static long getMedianDifference(java.util.TreeSet valueSet)
```

---

## getStandardDeviation

```
public static long getStandardDeviation(java.util.TreeSet valueSet)
```

---

## getLongestGap

```
public static long getLongestGap(java.util.TreeSet valueSet)
```

---

## overlap

```
public static boolean overlap(int start1,  
    int end1,  
    int start2,  
    int end2)
```

Check whether two numeric intervals overlap.

### Parameters:

start1 - The start1.  
end1 - The end1.  
start2 - The start2.  
end2 - The end2.

### Returns:

True, if the intervals overlap, false otherwise.

---

## calculateRMSE

```
public static double calculateRMSE(java.lang.String inputFile,  
    java.lang.String columnSeparator)
```

---

## calculateRMSE

```
public static double calculateRMSE(java.util.List values)
```

---

## calculateListSimilarity

```
public static ListSimilarity calculateListSimilarity(java.util.List list1,  
    java.util.List list2)
```

---

(continued from last page)

Calculate similarity of two lists of the same size.

**Parameters:**

list1 - The first list.  
list2 - The second list.

**Returns:**

The similarity of the two lists.

---

## calculateListSimilarity

```
public static ListSimilarity calculateListSimilarity(java.lang.String listFile,  
java.lang.String separator)
```

---

## performLinearRegression

```
public static double[] performLinearRegression(double[] x,  
double[] y)
```

Calculate the parameters for a regression line. A series of x and y must be given.  $y = \text{beta} * x + \text{alpha}$  TODO multiple regression model:  
[http://www.google.com/url?sa=t&source=web&cd=6&ved=0CC8QFjAF&url=http%3A%2F%2Fwww.bbn-school.org%2Fus%2Fmath%2Fap\\_stats%2Fproject\\_abstracts\\_folder%2Fproj\\_student\\_learning\\_folder%2Fmultiple\\_reg\\_ludlow.pps&ei=NQQ7TOHNCYacOPan6loK&usg=AFQjCNEybhIQVP2xwNGHEdYMgqNYelp1lQ&sig2=cwCNr1lvMv0PHwdwu\\_LIAQ](http://www.google.com/url?sa=t&source=web&cd=6&ved=0CC8QFjAF&url=http%3A%2F%2Fwww.bbn-school.org%2Fus%2Fmath%2Fap_stats%2Fproject_abstracts_folder%2Fproj_student_learning_folder%2Fmultiple_reg_ludlow.pps&ei=NQQ7TOHNCYacOPan6loK&usg=AFQjCNEybhIQVP2xwNGHEdYMgqNYelp1lQ&sig2=cwCNr1lvMv0PHwdwu_LIAQ), <http://www.stat.ufl.edu/~aa/sta6127/ch11.pdf> See [http://en.wikipedia.org/wiki/Simple\\_linear\\_regression](http://en.wikipedia.org/wiki/Simple_linear_regression) for an explanation.

**Parameters:**

x - A series of x values.  
y - A series of y values.

**Returns:**

The parameter alpha and beta for the regression line.



# tud.iir.helper

## Class StopWatch

```
java.lang.Object
|
|--tud.iir.helper.StopWatch
```

---

```
public class StopWatch
extends java.lang.Object
```

A simple stop watch for performance testing.

**Author:**

David Urbansky

---

## Constructors

### StopWatch

```
public StopWatch()
```

The StopWatch starts running right after object creation.

## Methods

### start

```
public void start()
```

Start/reset the stop watch.

---

### stop

```
public void stop()
```

Stop the stop watch.

---

### setCountDown

```
public void setCountDown(long countDown)
```

Set a count down in milliseconds.

---

### getCountDown

```
public long getCountDown()
```

Get the count down.

---

### timeIsUp

```
public boolean timeIsUp()
```

Check whether count down is up.

---

---

## getElapsedTime

```
public long getElapsedTime(boolean inSeconds)
```

Get the elapsed time.

**Parameters:**

`inSeconds` - If true, the elapsed time will be returned in seconds, otherwise in milliseconds.

**Returns:**

The elapsed time.

---

## getElapsedTime

```
public long getElapsedTime()
```

Get the elapsed time in milliseconds.

**Returns:**

The elapsed time.

---

## getElapsedTimeString

```
public java.lang.String getElapsedTimeString(boolean output)
```

Get the elapsed time as a string.

**Parameters:**

`output` - If true, the elapsed time will be printed to the console as well.

**Returns:**

The elapsed time as a string.

---

## getElapsedTimeString

```
public java.lang.String getElapsedTimeString()
```

Get the elapsed time as a string without console output.

**Returns:**

The elapsed time as a string.

---

## main

```
public static void main(java.lang.String[] args)
```

# tud.iir.helper

## Class StringHelper

```
java.lang.Object
└-- tud.iir.helper.StringHelper
```

---

```
public class StringHelper
extends java.lang.Object
```

The StringHelper adds string functionality.

**Author:**

David Urbansky, Martin Werner, Philipp Katz, Martin Gregor

---

## Constructors

### StringHelper

```
public StringHelper()
```

## Methods

### makeSafeName

```
public static java.lang.String makeSafeName(java.lang.String name)
```

In ontologies names can not have certain characters so they have to be changed.

**Parameters:**

`name` - The name.

**Returns:**

The safe name.

---

### toInt

```
public static java.lang.Integer toInt(java.lang.String text)
```

This function wraps the string to integer conversion in order to prevent the exception catching in other functions.

**Parameters:**

`text` - The text that is a number.

**Returns:**

The integer presentation of the text.

---

### toDouble

```
public static java.lang.Double toDouble(java.lang.String text)
```

This function wraps the string to double conversion in order to prevent the exception catching in other functions.

---

(continued from last page)

**Parameters:**

`text` - The text that is a number.

**Returns:**

The double presentation of the text.

---

## makeCamelCase

```
public static java.lang.String makeCamelCase(java.lang.String name,  
        boolean uppercaseFirst,  
        boolean toSingular)
```

Transform a name to a camel case variable name. For example: `car_speed` => `carSpeed` or `CarSpeed`

**Parameters:**

`name` - The name.

`uppercaseFirst` - If true, the first letter will be uppercase.

`toSingular` - If true, the last part is translated to its singular form.

**Returns:**

The camel cased name.

---

## makeCamelCase

```
public static java.lang.String makeCamelCase(java.lang.String name,  
        boolean uppercaseFirst)
```

Make camel case.

**Parameters:**

`name` - the name

`uppercaseFirst` - the uppercase first

**Returns:**

the string

---

## upperCaseFirstLetter

```
public static java.lang.String upperCaseFirstLetter(java.lang.String term)
```

Make first letter of word upper case.

**Parameters:**

`term` - The term.

**Returns:**

The term with an upper case first letter.

---

## lowerCaseFirstLetter

```
public static java.lang.String lowerCaseFirstLetter(java.lang.String term)
```

Make first letter of word lower case.

**Parameters:**

`term` - The term.

---

(continued from last page)

**Returns:**

The term with an lower case first letter.

---

## removeNumbering

```
public static java.lang.String removeNumbering(java.lang.String numberedText)
```

Replace number before a text. 1.1 Text => Text

**Parameters:**

`numberedText` - The text that possibly has numbers before it starts.

**Returns:**

The text without the numbers.

---

## makeViewName

```
public static java.lang.String makeViewName(java.lang.String name)
```

Make name for view.

**Parameters:**

`name` - The name.

**Returns:**

The view name.

---

## containsProperNoun

```
public static boolean containsProperNoun(java.lang.String searchString)
```

Check whether a given string contains a proper noun.

**Parameters:**

`searchString` - The search string.

**Returns:**

True if the string contains a proper noun, else false.

---

## containsNumber

```
public static boolean containsNumber(java.lang.String searchString)
```

Check whether a given string contains a numeric value.

**Parameters:**

`searchString` - The search string.

**Returns:**

True if the string contains a numeric value, else false.

---

## removeStopWords

```
public static java.lang.String removeStopWords(java.lang.String string)
```

Clean the given string from stop words, i.e. words that appear often but have no meaning itself.

**Parameters:**

(continued from last page)

`string` - The string.

**Returns:**

The string without the stop words.

---

## removeSpecialChars

```
public static java.lang.String removeSpecialChars(java.lang.String string)
```

Removes the special chars. TODO this does ... nothing? Marked this as deprecated -- Philipp.

**Parameters:**

`string` - the string

**Returns:**

the string

---

## removeNonAsciiCharacters

```
public static java.lang.String removeNonAsciiCharacters(java.lang.String string)
```

## removeBrackets

```
public static java.lang.String removeBrackets(java.lang.String bracketString)
```

Removes the brackets.

**Parameters:**

`bracketString` - the bracket string

**Returns:**

the string

---

## escapeForRegularExpression

```
public static java.lang.String escapeForRegularExpression(java.lang.String  
inputString)
```

Escape for regular expression.

**Parameters:**

`inputString` - the input string

**Returns:**

the string

---

## isBracket

```
public static boolean isBracket(char character)
```

Checks whether character is a bracket.

**Parameters:**

`character` - The character.

**Returns:**

True if character is a bracket, else false.

---

## isNumber

```
public static boolean isNumber(java.lang.Character ch)
```

Check if the string is a number.

**Parameters:**

ch - the ch

**Returns:**

True if string is number, else false.

---

## isNumber

```
public static boolean isNumber(java.lang.String string)
```

Checks if is number.

**Parameters:**

string - the string

**Returns:**

true, if is number

---

## isNumericExpression

```
public static boolean isNumericExpression(java.lang.String string)  
throws java.lang.NumberFormatException,  
       java.lang.OutOfMemoryError
```

Checks if is numeric expression.

**Parameters:**

string - the string

**Returns:**

true, if is numeric expression

**Throws:**

NumberFormatException - the number format exception

OutOfMemoryError - the out of memory error

---

## isTimeExpression

```
public static boolean isTimeExpression(java.lang.String string)
```

Checks if is time expression.

**Parameters:**

string - the string

**Returns:**

true, if is time expression

---

## isCompletelyUppercase

```
public static boolean isCompletelyUppercase(java.lang.String testString)
```

---

(continued from last page)

Checks if is completely uppercase.

**Parameters:**

`testString` - the test string

**Returns:**

true, if is completely uppercase

---

## startsWithUppercase

```
public static boolean startsWithUppercase(java.lang.String testString)
```

Starts uppercase.

**Parameters:**

`testString` - the test string

**Returns:**

true, if successful

---

## letterNumberCount

```
public static int letterNumberCount(java.lang.String string)
```

Letter number count.

**Parameters:**

`string` - the string

**Returns:**

the int

---

## numberCount

```
public static int numberCount(java.lang.String string)
```

---

## capitalizedWordCount

```
public static int capitalizedWordCount(java.lang.String string)
```

Capitalized word count.

**Parameters:**

`string` - the string

**Returns:**

the int

---

## isVowel

```
public static boolean isVowel(java.lang.Character inputCharacter)
```

Checks if is vowel.

**Parameters:**

`inputCharacter` - the input character

---



(continued from last page)

**Returns:**  
true, if is vowel

---

## trim

```
public static java.lang.String trim(java.lang.String string)
```

Remove unwanted characters from beginning and end of string.

**Parameters:**  
`string` - The string.

**Returns:**  
The trimmed string.

---

## trim

```
public static java.lang.String trim(java.lang.String inputString,  
    java.lang.String keepCharacters)
```

Trim.

**Parameters:**  
`inputString` - the input string  
`keepCharacters` - the keep characters

**Returns:**  
the string

---

## removeControlCharacters

```
public static java.lang.String removeControlCharacters(java.lang.String string)
```

---

## trim

```
public static java.util.HashSet trim(java.util.HashSet strings)
```

Trim.

**Parameters:**  
`strings` - the strings

**Returns:**  
the hash set

---

## makeContinuousText

```
public static java.lang.String makeContinuousText(java.lang.String text)
```

Remove tabs, line breaks and double spaces.

**Parameters:**  
`text` - The text to be cleaned.

**Returns:**  
The cleaned text.

---

## putArticleInFront

```
public static java.lang.String putArticleInFront(java.lang.String inputString)
```

Put article in front.

**Parameters:**

`inputString` - the input string

**Returns:**

the string

---

## countWords

```
public static int countWords(java.lang.String string)
```

Count number of words, words are separated by a blank " ".

**Parameters:**

`string` - The string.

**Returns:**

The number of words in the string.

---

## calculateSimilarity

```
public static double calculateSimilarity(java.lang.String string1,  
java.lang.String string2)
```

Calculate similarity.

**Parameters:**

`string1` - the string1

`string2` - the string2

**Returns:**

the double

---

## calculateSimilarity

```
public static double calculateSimilarity(java.lang.String string1,  
java.lang.String string2,  
boolean caseSensitive)
```

Calculate similarity.

**Parameters:**

`string1` - the string1

`string2` - the string2

`caseSensitive` - the case sensitive

**Returns:**

the double

---

(continued from last page)

## getLongestCommonString

```
public static java.lang.String getLongestCommonString(java.lang.String string1,  
    java.lang.String string2,  
    boolean caseSensitive,  
    boolean shiftString)
```

Get the longest common character chain two strings have in common.

### Parameters:

string1 - The first string.

string2 - The second string.

caseSensitive - True if the check should be case sensitive, false otherwise.

shiftString - If true, the shorter string will be shifted and checked against the longer string.

The longest common string of two strings is found regardless whether they start with the same characters. If true, ABCD and BBBCD have BCD in common, if false the longest common string is empty.

### Returns:

The longest common string.

---

## getArrayAsString

```
public static java.lang.String getArrayAsString(java.lang.String[] array)
```

Gets the array as string.

### Parameters:

array - the array

### Returns:

the array as string

---

## reverseString

```
public static java.lang.String reverseString(java.lang.String string)
```

Reverse a string. ABC => CBA.

### Parameters:

string - The string to be reversed.

### Returns:

The reversed string.

---

## concatMatchedString

```
public static java.lang.String concatMatchedString(java.lang.String inputString,  
    java.lang.String separator,  
    java.lang.String regularExpression)
```

Run a regular expression on a string and form a new string with the matched strings separated by the specified separator.

### Parameters:

inputString - The input string for the matching.

separator - The separator used to separate the matched strings.

regularExpression - The regular expression that is matched on the input string.

### Returns:

(continued from last page)

the string

---

## sha1

```
public static java.lang.String sha1(java.lang.String text)
```

Transform a given text into a 20 byte sha-1 encoded string.

**Parameters:**

`text` - The text to be encoded.

**Returns:**

The 20 byte (40 hexadecimal characters) string.

---

## encodeBase64

```
public static java.lang.String encodeBase64(java.lang.String string)
```

Encode base64.

**Parameters:**

`string` - the string

**Returns:**

the string

---

## decodeBase64

```
public static java.lang.String decodeBase64(java.lang.String string)
```

Decode base64.

**Parameters:**

`string` - the string

**Returns:**

the string

---

## getSubstringBetween

```
public static java.lang.String getSubstringBetween(java.lang.String string,  
    java.lang.String leftBorder,  
    java.lang.String rightBorder)
```

Get the substring between the given sequences.

**Parameters:**

`string` - The string where the substring belongs to.

`leftBorder` - The left border.

`rightBorder` - The right border.

**Returns:**

The substring between the two given strings or an empty string in case of an error.

---

## camelCaseToWords

```
public static java.lang.String camelCaseToWords(java.lang.String camelCasedString,  
    java.lang.String separator)
```

(continued from last page)

Transforms a CamelCased String into a split String.

**Parameters:**

`camelCasedString` - The String to split.

`separator` - The separator to insert between the camelCased fragments.

**Returns:**

The separated String.

---

## camelCaseToWords

```
public static java.lang.String camelCaseToWords(java.lang.String camelCasedString)
```

Transforms a CamelCased String into a space separated String. For example: `camelCaseString` is converted to `camel Case String`.

**Parameters:**

`camelCasedString` - The String to split.

**Returns:**

The separated String.

---

## urlDecode

```
public static java.lang.String urlDecode(java.lang.String url)
```

## urlEncode

```
public static java.lang.String urlEncode(java.lang.String string)
```

---

## removeFirstStringpart

```
public static java.lang.String[] removeFirstStringpart(java.lang.String string,  
java.lang.String regExp)
```

Looks for a regular expression in string. Removes found substring from source-string. Only the first found match will be deleted.

Return value consists of a two-field-array. First value is cleared string, second is removed substring.

**Parameters:**

`string` - to be cleared.

`regExp` - A regular expression.

**Returns:**

Cleared string and removed string in an array.

---

## main

```
public static void main(java.lang.String[] args)
```

The main method.

**Parameters:**

`args` - the arguments

---

---

## removeLastWhitespace

```
public static java.lang.String removeLastWhitespace(java.lang.String dateString)
```

Removes trailing whitespace at the end.

**Parameters:**

`dateString` - String to be cleared.

**Returns:**

Cleared string.

---

## removeDoubleWhitespaces

```
public static java.lang.String removeDoubleWhitespaces(java.lang.String text)
```

Replaces two or more trailing whitespaces by one.

**Parameters:**

`text`

**Returns:**

---

## countWhitespaces

```
public static int countWhitespaces(java.lang.String text)
```

Counts whitespace in a text.

**Parameters:**

`text`

**Returns:**

---

## tud.iir.helper

# Class StringInputStream

```
java.lang.Object
├-- java.io.InputStream
│   └-- tud.iir.helper.StringInputStream
```

All Implemented Interfaces:  
java.io.Closeable

---

```
public class StringInputStream
extends java.io.InputStream
```

---

## Constructors

### StringInputStream

```
public StringInputStream(java.lang.String text)
```

## Methods

### write

```
public void write(int b)
    throws java.io.IOException
```

---

### read

```
public int read()
    throws java.io.IOException
```

## tud.iir.helper Class StringOutputStream

```
java.lang.Object
  |
+- java.io.OutputStream
    |
+- tud.iir.helper.StringOutputStream
```

All Implemented Interfaces:  
java.io.Flushable, java.io.Closeable

---

```
public class StringOutputStream
extends java.io.OutputStream
```

---

### Constructors

#### **StringOutputStream**

```
public StringOutputStream()
```

### Methods

#### **write**

```
public void write(int b)
    throws java.io.IOException
```

---

#### **toString**

```
public java.lang.String toString()
```



## tud.iir.helper

# Class ThreadHelper

java.lang.Object

└─ tud.iir.helper.ThreadHelper

---

public class **ThreadHelper**  
extends java.lang.Object

---

## Constructors

### ThreadHelper

public **ThreadHelper**()

## Methods

### sleep

public static void **sleep**(int milliseconds)

# tud.iir.helper

## Class Tokenizer

```
java.lang.Object
└--tud.iir.helper.Tokenizer
```

---

```
public class Tokenizer
extends java.lang.Object
```

The Tokenizer tokenizes strings or creates chunks of that string.

**Author:**

David Urbansky

---

## Constructors

### Tokenizer

```
public Tokenizer()
```

## Methods

### tokenize

```
public static java.util.List tokenize(java.lang.String inputString)
```

Tokenize a given string.

**Parameters:**

`inputString` - The string to be tokenized.

**Returns:**

A list of tokens.

---

### calculateCharNGrams

```
public static java.util.Set calculateCharNGrams(java.lang.String string,
int n)
```

Calculate n-grams for a given string on a character level. The size of the set can be calculated as:  
Size = `stringLength` - `n` + 1

**Parameters:**

`string` - The string that the n-grams should be calculated for.

`n` - The number of characters for a gram.

**Returns:**

A set of n-grams.

---

(continued from last page)

---

## calculateWordNGrams

```
public static java.util.Set calculateWordNGrams(java.lang.String string,  
int n)
```

Calculate n-grams for a given string on a word level. The size of the set can be calculated as:  $\text{Size} = \text{numberOfWords} - n + 1$

### Parameters:

`string` - The string that the n-grams should be calculated for.  
`n` - The number of words for a gram.

### Returns:

A set of n-grams.

---

## calculateAllCharNGrams

```
public static java.util.Set calculateAllCharNGrams(java.lang.String string,  
int n1,  
int n2)
```

Calculate all n-grams for a string for different n on a character level. The size of the set can be calculated as:  $\text{Size} = \text{SUM}_n(n1, n2) (\text{stringLength} - n + 1)$

### Parameters:

`string` - The string the n-grams should be calculated for.  
`n1` - The smallest n-gram size.  
`n2` - The greatest n-gram size.

### Returns:

A set of n-grams.

---

## calculateAllWordNGrams

```
public static java.util.Set calculateAllWordNGrams(java.lang.String string,  
int n1,  
int n2)
```

Calculate all n-grams for a string for different n on a word level. The size of the set can be calculated as:  $\text{Size} = \text{SUM}_n(n1, n2) (\text{numberOfWords} - n + 1)$

### Parameters:

`string` - The string the n-grams should be calculated for.  
`n1` - The smallest n-gram size.  
`n2` - The greatest n-gram size.

### Returns:

A set of n-grams.

---

## getSentence

```
public static java.lang.String getSentence(java.lang.String string,  
int position)
```

Get the sentence that the specified position is in.

### Parameters:

`string` - The string.  
`position` - The position in the sentence.

---

(continued from last page)

**Returns:**

The whole sentence.

---

## getSentences

```
public static java.util.List getSentences(java.lang.String inputText)
```

Get a list of sentences of an input text. Also see <http://alias-i.com/lingpipe/demos/tutorial/sentences/read-me.html> for the LingPipe example.

**Parameters:**

`inputText` - An input text.

**Returns:**

A list with sentences.

---

## getPhraseFromBeginningOfSentence

```
public static java.lang.String getPhraseFromBeginningOfSentence(java.lang.String inputString)
```

Given a string, find the beginning of the sentence, e.g. "...now. Although, many of them" => "Although, many of them". consider !,?,. and : as end of sentence TODO control character after delimiter makes it end of sentence

**Parameters:**

`inputString` - the input string

**Returns:**

The phrase from the beginning of the sentence.

---

## getPhraseToEndOfSentence

```
public static java.lang.String getPhraseToEndOfSentence(java.lang.String string)
```

Given a string, find the end of the sentence, e.g. "Although, many of them (30.2%) are good. As long as" => "Although, many of them (30.2%) are good." consider !,?,. and : as end of sentence

**Parameters:**

`string` - The string.

**Returns:**

The phrase to the end of the sentence.

# tud.iir.helper

## Class TreeNode

java.lang.Object

└─ tud.iir.helper.TreeNode

All Implemented Interfaces:

java.io.Serializable

---

```
public class TreeNode
    extends java.lang.Object
    implements java.io.Serializable
```

A simple tree implementation. Identification of the nodes works via the labels. No tree node must have a label of another tree node.

**Author:**

David Urbansky

---

## Constructors

### TreeNode

```
public TreeNode(java.lang.String label)
```

---

### TreeNode

```
public TreeNode(java.lang.String label,
                java.util.HashMap children,
                TreeNode parent)
```

---

## Methods

### addNode

```
public boolean addNode(TreeNode tn)
```

Add a node as a child to the tree node.

**Parameters:**

tn - The tree node to add.

**Returns:**

True, if the node was not present, false otherwise.

---

### getLabel

```
public java.lang.String getLabel()
```

---

## setLabel

```
public void setLabel(java.lang.String label)
```

---

## getNode

```
public TreeNode getNode(java.lang.String label)
```

Get the node with the specified label that is somewhere below this node.

**Parameters:**

label - The label to search for.

**Returns:**

The sought TreeNode or null if it was not found.

---

## getChildren

```
public java.util.HashMap getChildren()
```

---

## setChildren

```
public void setChildren(java.util.HashMap children)
```

---

## getDescendants

```
public java.util.HashSet getDescendants()
```

---

## resetWeights

```
public void resetWeights()
```

Set all weights of the descendant nodes to 0.0.

---

## getParent

```
public TreeNode getParent()
```

---

## setParent

```
public void setParent(TreeNode parent)
```

---

## getRootPath

```
public java.util.ArrayList getRootPath()
```

(continued from last page)

Get all parent nodes until the root node is reached.

**Returns:**

An ordered list of parent nodes, ending with the root node.

---

## getLeafPath

```
public java.util.ArrayList getLeafPath()
```

Get all child nodes until the leaf node is reached. Follow the path of the highest weights.

**Returns:**

An ordered list of child nodes, ending with the leaf node.

---

## getFullPath

```
public java.util.ArrayList getFullPath()
```

Get an ordered list of all nodes before and after this node. Follow the children that have the highest weight.

**Returns:**

An ordered list of nodes from the leaf to the root.

---

## getValue

```
public java.lang.Object getValue()
```

---

## setValue

```
public void setValue(java.lang.Object value)
```

---

## getWeight

```
public double getWeight()
```

---

## setWeight

```
public void setWeight(double weight)
```

---

## toString

```
public java.lang.String toString()
```

---

## main

```
public static void main(java.lang.String[] args)
```

---

(continued from last page)

**Parameters:**

args



## tud.iir.helper

# Class WordNet

```
java.lang.Object
|
|--tud.iir.helper.WordNet
```

---

```
public class WordNet
extends java.lang.Object
```

---

## Constructors

### WordNet

```
public WordNet()
```

## Methods

### getSynonyms

```
public static java.lang.String[] getSynonyms(java.lang.String word,
                                             int number)
```

Return noun synonyms for the given word by looking it up in the WordNet database.

**Parameters:**

word - The word.  
number - The number.

**Returns:**

An array of synonyms.

---

### getSynonyms

```
public static java.lang.String[] getSynonyms(java.lang.String word,
                                             int number,
                                             boolean includeBaseWord)
```

---

### gerundToInfinitive

```
public static java.lang.String gerundToInfinitive(java.lang.String gerund)
```

(continued from last page)

Try to transform a gerund back to its infinitive form. The following code is very "ad hoc" and depends on the WordNet database. We simply try out different infinitive possibilities and check their occurrence and counts in WordNet. We assume that the occurrence with the highest count in WordNet is the correct infinitive form of the supplied gerund. Basically, there are three possibilities when transforming an infinitive to gerund:

1. think > thinking: Most simple variant by just appending the -ing suffix.
2. hit > hitting: The ending consonant is doubled, then -ing is appended.
3. take > tākeing: The -e is removed before appending -ing.

The problem when doing a reverse-transformation is, that we cannot know from the gerund form itself which of the above rules was applied (e. g. "thinking" vs. "taking"), so have to try out all three back-transformations.

**Parameters:**

gerund - the gerund to transform.

**Returns:**

infinitive form of the gerund, or the supplied word, if no transformation can be applied.

**See Also:**

[Forming Gerunds](#)  
[Wordnet](#)

---

## main

```
public static void main(java.lang.String[] args)
```

**Parameters:**

args

# tud.iir.helper

## Class WordTransformer

```
java.lang.Object
├-- tud.iir.helper.WordTransformer
```

---

```
public class WordTransformer
extends java.lang.Object
```

The WordTransformer transforms an input word. Currently it can transform English singular to plural and vice versa.

**Author:**  
David Urbansky, Philipp Katz

---

## Constructors

### WordTransformer

```
public WordTransformer()
```

## Methods

### wordToSingular

```
public static java.lang.String wordToSingular(java.lang.String pluralForm)
```

Transform an English plural word to its singular form.  
Rules: <http://www.englisch-hilfen.de/en/grammar/plural.htm>,  
[http://en.wikipedia.org/wiki/English\\_plural](http://en.wikipedia.org/wiki/English_plural)

**Parameters:**

pluralForm - The plural form of the word.

**Returns:**

The singular form of the word.

---

### wordToPlural

```
public static java.lang.String wordToPlural(java.lang.String singular)
```

Transform an English singular word to its plural form. rules:  
[http://owl.english.purdue.edu/handouts/grammar/g\\_spelnoun.html](http://owl.english.purdue.edu/handouts/grammar/g_spelnoun.html)

**Parameters:**

singular - The singular.

**Returns:**

The plural.

---

### main

```
public static void main(java.lang.String[] args)
```

---

(continued from last page)

# tud.iir.helper

## Class XPathHelper

```
java.lang.Object
└--tud.iir.helper.XPathHelper
```

---

```
public class XPathHelper
extends java.lang.Object
```

A helper to handle xPath.

**Author:**

David Urbansky, Philipp Katz, Martin Werner

---

## Constructors

### XPathHelper

```
public XPathHelper()
```

## Methods

### hasXMLNS

```
public static boolean hasXMLNS(org.w3c.dom.Document document)
```

Check whether document has a xhtml namespace declared.

**Parameters:**

`document` - The document.

**Returns:**

True if the document has a xhtml namespace declared, else false.

---

### addNamespaceToXPath

```
public static java.lang.String addNamespaceToXPath(org.w3c.dom.Document document,
    java.lang.String xpath)
```

Add the xhtml namespace to an XPath.

**Parameters:**

`document` - The document.

`xpath` - The XPath.

**Returns:**

The XPath with the namespace.

---

### addNamespaceToXPath

```
public static java.lang.String addNamespaceToXPath(java.lang.String xpath)
```

Add the xhtml namespace to an XPath in case it does not have it yet.

---

(continued from last page)

**Parameters:**

xPath - The XPath.

**Returns:**

The XPath with included xhtml namespace.

---

## getNodes

```
public static java.util.List getNodes(org.w3c.dom.Document document,  
    java.lang.String xpath)
```

Gets the nodes.

**Parameters:**

document - the document

xPath - the x path

**Returns:**

the nodes

---

## getNodes

```
public static java.util.List getNodes(org.w3c.dom.Node node,  
    java.lang.String xpath)
```

Gets the nodes.

**Parameters:**

node - the node

xPath - the x path

**Returns:**

the nodes

---

## getNode

```
public static org.w3c.dom.Node getNode(org.w3c.dom.Node node,  
    java.lang.String xpath)
```

Get a node by XPath.

**Parameters:**

node - The node where the XPath should be applied to.

xPath - The XPath.

**Returns:**

The node that the XPath points to.

---

## getNode

```
public static org.w3c.dom.Node getNode(org.w3c.dom.Document doc,  
    java.lang.String xpath)
```

Gets the node.

**Parameters:**

doc - the doc

xPath - the x path

---

(continued from last page)

**Returns:**  
the node

---

## getNodeById

```
public static org.w3c.dom.Node getNodeById(org.w3c.dom.Document document,  
                                             java.lang.String nodeId)
```

Gets the node by id.

**Parameters:**  
document - the document  
nodeId - the id

**Returns:**  
the node by id

---

## getChildNode

```
public static org.w3c.dom.Node getChildNode(org.w3c.dom.Node node,  
                                             java.lang.String xPath)
```

Get a child node by xPath.

**Parameters:**  
node - The parent node under which the sought node must descend.  
xPath - The xPath that points to a node.

**Returns:**  
A node that matches the xPath and descends from the given node.

---

## getChildNodes

```
public static java.util.List getChildNodes(org.w3c.dom.Node node,  
                                             java.lang.String xPath)
```

Gets the child nodes.

**Parameters:**  
node - the node  
xPath - the x path

**Returns:**  
the child nodes

---

## getChildNodes

```
public static java.util.List getChildNodes(org.w3c.dom.Node node)
```

Gets the child nodes.

**Parameters:**  
node - the (parent)node

**Returns:**  
the childNodes

---

(continued from last page)

## convertNodeToString

```
public static java.lang.String convertNodeToString(org.w3c.dom.Node node)
```

Convert a node and his children to string.

**Parameters:**

node - the node

**Returns:**

the node as string

---

## getPreviousSiblings

```
public static java.util.List getPreviousSiblings(org.w3c.dom.Node node)
```

Gets the previous sibling nodes of a node.

**Parameters:**

node - the node

**Returns:**

the previous siblings



---

Package

**tud.iir.helper.shingling**

# tud.iir.helper.shingling

## Class BitPermutations

java.lang.Object

└--tud.iir.helper.shingling.BitPermutations

---

public class **BitPermutations**  
extends java.lang.Object

From [http://it.toolbox.com/wiki/index.php/Perform\\_bitwise\\_permutation\\_using\\_Java](http://it.toolbox.com/wiki/index.php/Perform_bitwise_permutation_using_Java)

**Author:**  
Philipp Katz

---

## Constructors

### BitPermutations

```
public BitPermutations()
```

## Methods

### perm

```
public static long perm(long b)
```

---

### showBits

```
public static void showBits(long l)
```

---

### main

```
public static void main(java.lang.String[] args)
```

# tud.iir.helper.shingling Class Shingles

java.lang.Object

└--tud.iir.helper.shingling.Shingles

---

public class **Shingles**  
extends java.lang.Object

Simplified Shingle implementation to detect near-duplicate documents. All documents added are stored with an ID and their corresponding sketches in an index, to allow lookups for duplicates.

<http://www.ida.liu.se/~TDDC03/oldprojects/2005/final-projects/prj10.pdf>

<http://www.cs.princeton.edu/courses/archive/spr05/cos598E/bib/Princeton.pdf> <http://phpir.com/shingling-near-duplicate-detection> <http://www.std.org/~msm/common/clustering.html> [http://isabel-drost.de/projects/tuberlin/imsem2010/dups\\_paper\\_2010.pdf](http://isabel-drost.de/projects/tuberlin/imsem2010/dups_paper_2010.pdf)

<http://codingplayground.blogspot.com/2008/06/shingling-and-text-clustering.html> TODO useful preprocessing steps? make lower case, remove punctuation, remove duplicate white space?

**Author:**

Philipp Katz

---

## Fields

### DEFAULT\_N\_GRAM\_LENGTH

public static final int **DEFAULT\_N\_GRAM\_LENGTH**

Constant value: 3

---

### DEFAULT\_SKETCH\_SIZE

public static final int **DEFAULT\_SKETCH\_SIZE**

Constant value: 200

---

### DEFAULT\_SIMILARITY\_THRESHOLD

public static final float **DEFAULT\_SIMILARITY\_THRESHOLD**

Constant value: 0.1

---

## Constructors

### Shingles

public **Shingles**()

---

### Shingles

public **Shingles**([ShinglesIndex](#) index)

---

(continued from last page)

Initialize with a specific [ShinglesIndex](#) implementation.

**Parameters:**

index

## Methods

### addDocument

```
public boolean addDocument(int documentId,  
    java.lang.String documentContent)
```

Add a document's content to the shingle collection. A document is uniquely represented by an ID.

**Parameters:**

documentId

documentContent

**Returns:**

true, if document was similar/duplicate.

---

### addFile

```
public boolean addFile(java.lang.String filePath)
```

Add a file to the shingle collection.

**Parameters:**

filePath

**Returns:**

true, if document was similar/duplicate.

---

### addDocumentsFromFile

```
public java.util.Collection addDocumentsFromFile(java.lang.String filePath)
```

Adds multiple documents from one file to the shingle collection. Each document is on its own line. Line number is the document's ID.

**Parameters:**

filePath

**Returns:**

list of document IDs which already have similar/identical documents in the collection.

---

### getSimilarDocuments

```
public java.util.Map getSimilarDocuments()
```

Get a map with similar documents. E.g. [1 -> 5, 6, 10]

**Returns:**

---

### getSimilarityReport

```
public java.lang.String getSimilarityReport()
```

---

---

## getnGramLength

```
public int getnGramLength()
```

---

## setnGramLength

```
public void setnGramLength(int shingleLength)
```

Set length of shingles/n-grams.

**Parameters:**

nGramLength

---

## getSketchSize

```
public int getSketchSize()
```

---

## setSketchSize

```
public void setSketchSize(int sketchSize)
```

Set size of the sketch,

**Parameters:**

sketchSize

---

## getSimilarityThreshold

```
public float getSimilarityThreshold()
```

---

## setSimilarityThreshold

```
public void setSimilarityThreshold(float similarityThreshold)
```

Set threshold when two documents are considered "near duplicates".

**Parameters:**

similarityThreshold

---

## jaccardDistance

```
public static float jaccardDistance(java.util.Set s1,  
    java.util.Set s2)
```

Calculate Jaccard distance. The bigger the result, the more dissimilar are the two sets.  
[http://en.wikipedia.org/wiki/Jaccard\\_index](http://en.wikipedia.org/wiki/Jaccard_index)

**Parameters:**

s1

s2

(continued from last page)

**Returns:**

value between inclusive 0 and 1. Bigger value means more dissimilar.

---

## getMinN

```
public static java.util.Set getMinN(java.util.Set set,  
                                     int n)
```

Returns the "minimum" n items of the specified set, which are determined via their `Comparable.compareTo(T)` methods.

**Parameters:**

`set` - the input Set.

`n` - the number of minimum elements to return.

**Returns:**

the n minimum elements of the set.

---

## main

```
public static void main(java.lang.String[] args)  
    throws java.lang.Exception
```

---

## saveIndex

```
public void saveIndex()
```

**See Also:**

[ShinglesIndex.saveIndex\(\)](#)

# tud.iir.helper.shingling Interface ShinglesIndex

All Known Implementing Classes:

[ShinglesIndexBaseImpl](#), [ShinglesIndexTracer](#)

---

public interface **ShinglesIndex**  
extends

Defines an Index to store Shingle specific data. The model includes documents represented by a unique ID and their sketches, which are sets of hashed n-grams. The interface allows the lookup of documents based on their sketch or hashes und the lookup of sketches for specific documents. Further we keep a references between similar/identical documents.

**Author:**

Philipp Katz

---

## Methods

### setIndexName

```
public void setIndexName(java.lang.String name)
```

Set the name of this index. For instance, we might have different document collections which use their own indices.

**Parameters:**

name

---

### getIndexName

```
public java.lang.String getIndexName()
```

Get the name of this index.

**Returns:**

---

### openIndex

```
public void openIndex()
```

Open the index for usage. This must be the first call to the index instance.

---

### saveIndex

```
public void saveIndex()
```

Save the index, if necessary.

---

### deleteIndex

```
public void deleteIndex()
```

Delete the index, e.g. its corresponding files. This is intended for clean up after unit testing.

---

---

## addDocument

```
public void addDocument(int documentId,  
    java.util.Set sketch)
```

Add a document which is represented by an ID and its sketch (aka. set of hashes) to the index.

**Parameters:**

```
    documentId  
    sketch
```

---

## getDocumentsForHash

```
public java.util.Set getDocumentsForHash(long hash)
```

Get all document IDs for the specified hash.

**Parameters:**

```
    hash
```

**Returns:**

---

## getDocumentsForSketch

```
public java.util.Map getDocumentsForSketch(java.util.Set sketch)
```

**Deprecated.** *this is generally slow.*

Get all documents for the specified sketch. This will return all documents which contain at least one hash from the sketch.

**Parameters:**

```
    sketch
```

**Returns:**

---

## getSketchForDocument

```
public java.util.Set getSketchForDocument(int documentId)
```

Get the sketch for a stored document.

**Parameters:**

```
    documentId
```

**Returns:**

---

## getNumberOfDocuments

```
public int getNumberOfDocuments()
```

Get count of stored documents.

**Returns:**

---



---

## getSimilarDocuments

```
public java.util.Set getSimilarDocuments(int documentId)
```

Get similar documents for a specific document.

**Parameters:**

documentId

**Returns:**

---

## addDocumentSimilarity

```
public void addDocumentSimilarity(int masterDocumentId,  
    int similarDocumentId)
```

Add a similarity relation between two documents.

**Parameters:**

masterDocumentId

similarDocumentId

---

## getSimilarDocuments

```
public java.util.Map getSimilarDocuments()
```

Get a map of all similar documents.

**Returns:**

# tud.iir.helper.shingling Class ShinglesIndexBaseImpl

java.lang.Object

└─tud.iir.helper.shingling.ShinglesIndexBaseImpl

All Implemented Interfaces:

[ShinglesIndex](#)

Direct Known Subclasses:

[ShinglesIndexH2](#), [ShinglesIndexJava](#), [ShinglesIndexJDBM](#), [ShinglesIndexWB](#)

---

public abstract class **ShinglesIndexBaseImpl**  
extends java.lang.Object  
implements [ShinglesIndex](#)

Base ShinglesIndex implementation, with common functionality. [openIndex\(\)](#) and [saveIndex\(\)](#) can be overridden by subclasses as necessary.

**Author:**

Philipp Katz

---

## Fields

### INDEX\_FILE\_BASE\_PATH

public static final java.lang.String **INDEX\_FILE\_BASE\_PATH**

default directory where to store serialized shingles.  
Constant value: `data/models/shingles/`

## Constructors

### ShinglesIndexBaseImpl

public **ShinglesIndexBaseImpl**()

## Methods

### getIndexName

public java.lang.String **getIndexName**()

---

### setIndexName

public void **setIndexName**(java.lang.String indexName)

---

(continued from last page)

## openIndex

```
public void openIndex()
```

---

## saveIndex

```
public void saveIndex()
```

---

## deleteIndex

```
public void deleteIndex()
```

---

## getDocumentsForSketch

```
public java.util.Map getDocumentsForSketch(java.util.Set sketch)
```

# tud.iir.helper.shingling Class ShinglesIndexH2

java.lang.Object

└- [tud.iir.helper.shingling.ShinglesIndexBaseImpl](#)  
└- **tud.iir.helper.shingling.ShinglesIndexH2**

All Implemented Interfaces:  
[ShinglesIndex](#)

---

```
public class ShinglesIndexH2  
extends ShinglesIndexBaseImpl
```

Author:  
Philipp Katz

---

## Constructors

### ShinglesIndexH2

```
public ShinglesIndexH2()
```

---

## Methods

### addDocument

```
public void addDocument(int documentId,  
    java.util.Set sketch)
```

---

### getDocumentsForHash

```
public java.util.Set getDocumentsForHash(long hash)
```

---

### getDocumentsForSketch

```
public java.util.Map getDocumentsForSketch(java.util.Set sketch)
```

---

### getSketchForDocument

```
public java.util.Set getSketchForDocument(int documentId)
```

---

(continued from last page)

## addDocumentSimilarity

```
public void addDocumentSimilarity(int masterDocumentId,  
    int similarDocumentId)
```

---

## getSimilarDocuments

```
public java.util.Map getSimilarDocuments()
```

---

## getSimilarDocuments

```
public java.util.Set getSimilarDocuments(int documentId)
```

---

## getNumberOfDocuments

```
public int getNumberOfDocuments()
```

---

## deleteIndex

```
public void deleteIndex()
```

---

## main

```
public static void main(java.lang.String[] args)
```

---

# tud.iir.helper.shingling Class ShinglesIndexJava

java.lang.Object

└- [tud.iir.helper.shingling.ShinglesIndexBaseImpl](#)  
└- **tud.iir.helper.shingling.ShinglesIndexJava**

All Implemented Interfaces:  
[ShinglesIndex](#)

---

public class **ShinglesIndexJava**  
extends [ShinglesIndexBaseImpl](#)

Shingle index with in-memory Java Object graph. Persistence is achieved via Java serialization.

**Author:**  
Philipp Katz

---

## Constructors

### ShinglesIndexJava

```
public ShinglesIndexJava()
```

---

## Methods

### addDocument

```
public void addDocument(int documentId,  
    java.util.Set sketch)
```

---

### getDocumentsForHash

```
public java.util.Set getDocumentsForHash(long hash)
```

---

### getSketchForDocument

```
public java.util.Set getSketchForDocument(int documentId)
```

---

### addDocumentSimilarity

```
public void addDocumentSimilarity(int masterDocumentId,  
    int similarDocumentId)
```

---

## getSimilarDocuments

```
public java.util.Map getSimilarDocuments()
```

---

## getSimilarDocuments

```
public java.util.Set getSimilarDocuments(int documentId)
```

---

## getNumberOfDocuments

```
public int getNumberOfDocuments()
```

---

## openIndex

```
public void openIndex()
```

---

## saveIndex

```
public void saveIndex()
```

---

## deleteIndex

```
public void deleteIndex()
```

---

## main

```
public static void main(java.lang.String[] args)
```

# tud.iir.helper.shingling Class ShinglesIndexJDBM

java.lang.Object

└-[tud.iir.helper.shingling.ShinglesIndexBaseImpl](#)  
└-tud.iir.helper.shingling.ShinglesIndexJDBM

All Implemented Interfaces:  
[ShinglesIndex](#)

---

public class **ShinglesIndexJDBM**  
extends [ShinglesIndexBaseImpl](#)

Implementation of a ShinglesIndex which uses B+Trees via JDBM. <http://jdbm.sourceforge.net/>  
<http://www.antonioshome.net/blog/2006/20060224-1.php>  
<http://directory.apache.org/apacheds/1.5/table-and-cursor-implementations.html>

**Author:**  
Philipp Katz

---

## Constructors

### ShinglesIndexJDBM

public **ShinglesIndexJDBM**()

## Methods

### openIndex

public void **openIndex**()

For testing purposes, will use a temp. file with random name as index.

---

### deleteIndex

public void **deleteIndex**()

---

### addDocument

public void **addDocument**(int documentId,  
java.util.Set sketch)

---

### addDocumentSimilarity

public void **addDocumentsSimilarity**(int masterDocumentId,  
int similarDocumentId)

---



(continued from last page)

---

## getDocumentsForHash

```
public java.util.Set getDocumentsForHash(long hash)
```

---

## getNumberOfDocuments

```
public int getNumberOfDocuments()
```

---

## getSketchForDocument

```
public java.util.Set getSketchForDocument(int documentId)
```

---

## getSimilarDocuments

```
public java.util.Set getSimilarDocuments(int documentId)
```

---

## getSimilarDocuments

```
public java.util.Map getSimilarDocuments()
```

# tud.iir.helper.shingling

## Class ShinglesIndexTracer

java.lang.Object

└--tud.iir.helper.shingling.ShinglesIndexTracer

All Implemented Interfaces:

[ShinglesIndex](#)

---

```
public class ShinglesIndexTracer
    extends java.lang.Object
    implements ShinglesIndex
```

Decorator to allow performance testing.

**Author:**

Philipp Katz

---

## Constructors

### ShinglesIndexTracer

```
public ShinglesIndexTracer(ShinglesIndex profiled)
```

---

## Methods

### addDocument

```
public void addDocument(int documentId,
    java.util.Set sketch)
```

---

### addDocumentSimilarity

```
public void addDocumentSimilarity(int masterDocumentId,
    int similarDocumentId)
```

---

### getDocumentsForHash

```
public java.util.Set getDocumentsForHash(long hash)
```

---

### getDocumentsForSketch

```
public java.util.Map getDocumentsForSketch(java.util.Set sketch)
```

---

## getNumberOfDocuments

```
public int getNumberOfDocuments()
```

---

---

## getSimilarDocuments

```
public java.util.Set getSimilarDocuments(int documentId)
```

---

---

## getSimilarDocuments

```
public java.util.Map getSimilarDocuments()
```

---

---

## getSketchForDocument

```
public java.util.Set getSketchForDocument(int documentId)
```

---

---

## getTraceResult

```
public java.lang.String getTraceResult()
```

---

---

## deleteIndex

```
public void deleteIndex()
```

---

---

## getIndexName

```
public java.lang.String getIndexName()
```

---

---

## openIndex

```
public void openIndex()
```

---

---

## saveIndex

```
public void saveIndex()
```

---

---

## setIndexName

```
public void setIndexName(java.lang.String name)
```

---

(continued from last page)

# tud.iir.helper.shingling Class ShinglesIndexWB

java.lang.Object

└-[tud.iir.helper.shingling.ShinglesIndexBaseImpl](#)  
└-tud.iir.helper.shingling.ShinglesIndexWB

All Implemented Interfaces:  
[ShinglesIndex](#)

---

public class **ShinglesIndexWB**  
extends [ShinglesIndexBaseImpl](#)

ShinglesIndex implementation using "WB B-Tree Database". The API is plain shocking and seems to be ported directly from C. TODO this does not work if we have non continuous IDs ... like 1, 2, 9, 17, ...  
<http://people.csail.mit.edu/jaffer/WB>

Author:  
Philipp Katz

---

## Constructors

### ShinglesIndexWB

public **ShinglesIndexWB**()

## Methods

### openIndex

public void **openIndex**()

---

### deleteIndex

public void **deleteIndex**()

---

### addDocument

public void **addDocument**(int documentId,  
java.util.Set sketch)

---

### addDocumentSimilarity

public void **addDocumentsSimilarity**(int masterDocumentId,  
int similarDocumentId)

(continued from last page)

---

## getDocumentsForHash

```
public java.util.Set getDocumentsForHash(long hash)
```

---

## getNumberOfDocuments

```
public int getNumberOfDocuments()
```

---

## getSimilarDocuments

```
public java.util.Set getSimilarDocuments(int documentId)
```

---

## getSimilarDocuments

```
public java.util.Map getSimilarDocuments()
```

---

## getSketchForDocument

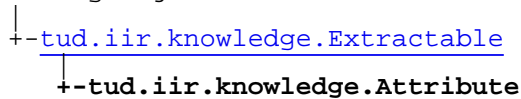
```
public java.util.Set getSketchForDocument(int documentId)
```

---

Package  
**tud.iir.knowledge**

# tud.iir.knowledge Class Attribute

java.lang.Object



All Implemented Interfaces:  
java.io.Serializable

```
public class Attribute
extends Extractable
```

The knowledge unit attribute.

**Author:**  
David Urbansky

## Fields

### VALUE\_NUMERIC

```
public static final int VALUE_NUMERIC
```

Constant value: 1

### VALUE\_STRING

```
public static final int VALUE_STRING
```

Constant value: 2

### VALUE\_DATE

```
public static final int VALUE_DATE
```

Constant value: 3

### VALUE\_BOOLEAN

```
public static final int VALUE_BOOLEAN
```

Constant value: 4

### VALUE\_IMAGE

```
public static final int VALUE_IMAGE
```

Constant value: 5



---

## VALUE\_VIDEO

```
public static final int VALUE_VIDEO
```

Constant value: 6

---

## VALUE\_AUDIO

```
public static final int VALUE_AUDIO
```

Constant value: 7

---

## VALUE\_MIXED

```
public static final int VALUE_MIXED
```

Constant value: 8

---

## VALUE\_URI

```
public static final int VALUE_URI
```

Constant value: 9

---

## Constructors

### Attribute

```
public Attribute(java.lang.String name,  
                 int valueType,  
                 Concept concept)
```

---

### Attribute

```
public Attribute(java.lang.String name,  
                 int valueType,  
                 Concept concept,  
                 java.lang.String regExp)
```

---

### Attribute

```
public Attribute(java.lang.String name,  
                 int valueType,  
                 Concept concept,  
                 double trust)
```

## Methods

(continued from last page)

## getValueTypeByName

```
public static int getValueTypeByName(java.lang.String name)
```

---

## hasSynonym

```
public boolean hasSynonym(java.lang.String name)
```

---

## getSynonyms

```
public java.util.HashSet getSynonyms()
```

---

## getSynonymsToString

```
public java.lang.String getSynonymsToString()
```

---

## setSynonyms

```
public void setSynonyms(java.util.HashSet synonyms)
```

---

## addSynonym

```
public void addSynonym(java.lang.String synonym)
```

---

## guessValueType

```
public static int guessValueType(java.lang.String factString,  
    int mode)
```

Take a fact string of unknown value type and try to guess which type it is.

### Parameters:

`factString` - The unknown fact string.

`mode` - If 1 a numeric data type is assumed if fact string contains a number, in mode 2 fact string must start with a number.

### Returns:

The guessed value type.

---

## getValueType

```
public int getValueType()
```

---

(continued from last page)

---

## getValueTypeName

```
public java.lang.String getValueTypeName()
```

---

## getValueTypeXSD

```
public Resource getValueTypeXSD()
```

---

## setValueType

```
public void setValueType(int valueType)
```

---

## getSaveType

```
public java.lang.String getSaveType()
```

---

## setSaveType

```
public void setSaveType(java.lang.String saveType)
```

---

## getRegExp

```
public java.lang.String getRegExp()
```

---

## setRegExp

```
public void setRegExp(java.lang.String regExp)
```

---

## getConcept

```
public Concept getConcept()
```

---

## getPredefinedSources

```
public java.util.HashSet getPredefinedSources()
```

---

## setPredefinedSources

```
public void setPredefinedSources(java.util.HashSet predefinedSources)
```

---

---

## addPredefinedSource

```
public void addPredefinedSource(Source source)
```

---

## getValueCount

```
public int getValueCount()
```

---

## setValueCount

```
public void setValueCount(int valueCount)
```

---

## isExtracted

```
public boolean isExtracted()
```

---

## toString

```
public java.lang.String toString()
```

---

## getNewName

```
public java.lang.String getNewName()
```

---

## getSafeNewName

```
public java.lang.String getSafeNewName()
```

---

## setNewName

```
public void setNewName(java.lang.String newName)
```

---

## getRangeString

```
public java.lang.String getRangeString()
```

---

(continued from last page)

## getAttributeRanges

```
public java.util.HashSet getAttributeRanges()
```

---

## getAttributeRangesToDelete

```
public java.util.HashSet getAttributeRangesToDelete()
```

---

## addRangeNodeDummies

```
public void addRangeNodeDummies(java.lang.String rangeConceptName,  
    java.lang.String rangeType)
```

---

## addRangeValue

```
public void addRangeValue(AttributeRange rangeValueItem)
```

---

## getRange

```
public AttributeRange getRange(java.lang.String conceptName)
```

---

## removeRange

```
public void removeRange(AttributeRange range)
```

---

## addRangeValue

```
public boolean addRangeValue(java.lang.String rangeValueString,  
    java.lang.String rangeConceptName)
```

---

## removeRangeValue

```
public void removeRangeValue(java.lang.String rangeValue,  
    java.lang.String rangeConceptName)
```

---

## clearRangeValues

```
public void clearRangeValues()
```

---

(continued from last page)

## getNewSynonyms

```
public java.util.HashSet getNewSynonyms()
```

---

## setNewSynonyms

```
public void setNewSynonyms(java.util.HashSet newSynonyms)
```

---

## hasNewSynonyms

```
public boolean hasNewSynonyms()
```

# tud.iir.knowledge

## Class AttributeRange

java.lang.Object

└--tud.iir.knowledge.AttributeRange

public class **AttributeRange**  
extends java.lang.Object

## Fields

### UNIT\_UNITLESS

public static final int **UNIT\_UNITLESS**

Constant value: 0

### UNIT\_TIME

public static final int **UNIT\_TIME**

Constant value: 1

### UNIT\_DIGITAL

public static final int **UNIT\_DIGITAL**

Constant value: 2

### UNIT\_FREQUENCY

public static final int **UNIT\_FREQUENCY**

Constant value: 3

### UNIT\_LENGTH

public static final int **UNIT\_LENGTH**

Constant value: 4

### UNIT\_WEIGHT

public static final int **UNIT\_WEIGHT**

Constant value: 5

---

## RANGETYPE\_MINMAX

```
public static final java.lang.String RANGETYPE_MINMAX
```

Constant value: **MINMAX**

---

## RANGETYPE\_POSS

```
public static final java.lang.String RANGETYPE_POSS
```

Constant value: **POSS**

---

## Constructors

### AttributeRange

```
public AttributeRange(java.lang.String rangeConcept)
```

## Methods

### hasPossValue

```
public boolean hasPossValue()
```

---

### hasMinValue

```
public boolean hasMinValue()
```

---

### hasMaxValue

```
public boolean hasMaxValue()
```

---

### getRangeMinValue

```
public java.lang.String getRangeMinValue()
```

---

### getRangeMaxValue

```
public java.lang.String getRangeMaxValue()
```

---

### getRangePossValues

```
public java.util.ArrayList getRangePossValues()
```

---



---

## getRangeString

```
public java.lang.String getRangeString()
```

---

## addRangeValue

```
public boolean addRangeValue(java.lang.String rangeValue,  
    int valueType)
```

---

## removeRangeValue

```
public void removeRangeValue(java.lang.String rangeValueString)
```

---

## clearRangeValues

```
public void clearRangeValues()
```

---

## getRangeType

```
public java.lang.String getRangeType()
```

---

## setRangeType

```
public void setRangeType(java.lang.String rangeType)
```

---

## getRangeConcept

```
public java.lang.String getRangeConcept()
```

# tud.iir.knowledge

## Class Concept

java.lang.Object

└-- tud.iir.knowledge.Concept

All Implemented Interfaces:

java.io.Serializable

---

```
public class Concept
  extends java.lang.Object
  implements java.io.Serializable
```

The knowledge unit concept.

**Author:**

David Urbansky

---

## Constructors

### Concept

```
public Concept(java.lang.String name)
```

---

### Concept

```
public Concept(java.lang.String name,
               KnowledgeManager knowledgeManager)
```

---

## Methods

### getID

```
public int getID()
```

---

### setID

```
public void setID(int id)
```

---

### getSuperClass

```
public java.lang.String getSuperClass()
```

---

(continued from last page)

## setSuperClass

```
public void setSuperClass(java.lang.String superClass)
```

---

## getNewSuperClass

```
public java.lang.String getNewSuperClass()
```

---

## setNewSuperClass

```
public void setNewSuperClass(java.lang.String newSuperClass)
```

---

## getName

```
public java.lang.String getName()
```

---

## getSafeName

```
public java.lang.String getSafeName()
```

---

## hasSynonym

```
public boolean hasSynonym(java.lang.String name)
```

---

## setName

```
public void setName(java.lang.String name)
```

---

## getNewName

```
public java.lang.String getNewName()
```

---

## getSafeNewName

```
public java.lang.String getSafeNewName()
```

---

## setNewName

```
public void setNewName(java.lang.String newName)
```

---

---

## getSynonyms

```
public java.util.HashSet getSynonyms()
```

---

## getNewSynonyms

```
public java.util.HashSet getNewSynonyms()
```

---

## setNewSynonyms

```
public void setNewSynonyms(java.util.HashSet newSynonyms)
```

---

## hasNewSynonyms

```
public boolean hasNewSynonyms()
```

---

## getSynonymsToString

```
public java.lang.String getSynonymsToString()
```

---

## setSynonyms

```
public void setSynonyms(java.util.HashSet synonyms)
```

---

## addSynonym

```
public void addSynonym(java.lang.String synonym)
```

---

## getKnowledgeManager

```
public KnowledgeManager getKnowledgeManager()
```

---

## setKnowledgeManager

```
public void setKnowledgeManager(KnowledgeManager knowledgeManager)
```

---

(continued from last page)

## getLastSearched

```
public java.util.Date getLastSearched()
```

---

## setLastSearched

```
public void setLastSearched(java.util.Date lastSearched)
```

---

## getAttributes

```
public java.util.HashSet getAttributes()
```

---

## getAttributes

```
public java.util.HashSet getAttributes(boolean onlyManuallyAdded)
```

---

## getAttributesToDelete

```
public java.util.HashSet getAttributesToDelete()
```

---

## getAttributesAsList

```
public java.util.ArrayList getAttributesAsList(boolean onlyManuallyAdded)
```

---

## getAttributeNames

```
public java.util.HashSet getAttributeNames()
```

---

## setAttributes

```
public void setAttributes(java.util.HashSet attributes)
```

---

## addAttribute

```
public boolean addAttribute(Attribute attribute)
```

---

## hasAttribute

```
public boolean hasAttribute(java.lang.String attributeName)
```

---

---

## hasAttribute

```
public boolean hasAttribute(java.lang.String attributeName,  
    boolean onlyManuallyAdded)
```

---

## getAttribute

```
public Attribute getAttribute(java.lang.String attributeName)
```

---

## getAttribute

```
public Attribute getAttribute(java.lang.String attributeName,  
    boolean useSynonyms)
```

---

## getAttribute

```
public Attribute getAttribute(int attributeId)
```

---

## removeAttribute

```
public boolean removeAttribute(int attributeId)
```

---

## getEntities

```
public java.util.ArrayList getEntities()
```

---

## getEntitiesByTrust

```
public java.util.ArrayList getEntitiesByTrust()
```

---

## getEntitiesByDate

```
public java.util.ArrayList getEntitiesByDate()
```

---

## clearEntities

```
public void clearEntities()
```

## setEntities

```
public void setEntities(java.util.ArrayList entities)
```

---

## addEntity

```
public void addEntity(Entity entity)
```

---

## hasEntity

```
public boolean hasEntity(java.lang.String entityName)
```

---

## getEntity

```
public Entity getEntity(java.lang.String entityName)
```

---

## loadEntities

```
public void loadEntities(boolean continueFromLastExtraction)
```

Load entities for the concept from the rdb. Load oldest (lastSearched) first.

**Parameters:**

`continueFromLastExtraction` - If true, the counter is set to the last extraction and it will be continued from there.

---

## toString

```
public final java.lang.String toString()
```

# tud.iir.knowledge

## Class Entity

java.lang.Object

└- [tud.iir.knowledge.Extractable](#)  
└- **tud.iir.knowledge.Entity**

All Implemented Interfaces:  
java.io.Serializable

public class **Entity**  
extends [Extractable](#)

The knowledge unit entity.

**Author:**  
David Urbansky

## Constructors

### Entity

```
public Entity(java.lang.String name,  
              Concept concept,  
              boolean initial)
```

### Entity

```
public Entity(java.lang.String name,  
              Concept concept,  
              double trust)
```

### Entity

```
public Entity(java.lang.String name,  
              Concept concept)
```

### Entity

```
public Entity(java.lang.String name)
```

## Methods

### getConcept

```
public Concept getConcept()
```



---

## setConcept

```
public void setConcept(Concept concept)
```

---

## getSnippets

```
public java.util.ArrayList getSnippets()
```

---

## addSnippets

```
public void addSnippets(java.util.List snippets)
```

---

## getFacts

```
public java.util.ArrayList getFacts()
```

---

## getFactForAttribute

```
public Fact getFactForAttribute(Attribute attribute)
```

---

## setFacts

```
public void setFacts(java.util.ArrayList facts)
```

---

## addFactForBenchmark

```
public void addFactForBenchmark(Fact fact,  
                                FactValue factValue)
```

---

## addFactAndValue

```
public void addFactAndValue(Fact fact,  
                             FactValue factValue)
```

---

## getNumberOfExtractions

```
public int getNumberOfExtractions(int extractionType)
```

## isInitial

```
public boolean isInitial()
```

---

## setInitial

```
public void setInitial(boolean initial)
```

---

## getExtractionCount

```
public int getExtractionCount()
```

Return the number of times the entity has been extracted.

**Returns:**

Number of times the entity has been extracted.

---

## getExtractionTypeCount

```
public int getExtractionTypeCount()
```

Return the distinct number extraction types used to extract the entity.

**Returns:**

Number of times the entity has been extracted.

---

## getExtractionTypes

```
public java.util.HashSet getExtractionTypes()
```

Return a set of the extraction types used to extract the entity.

**Returns:**

Set of extractionTypes used to extract the entity:

---

## isCorrect

```
public boolean isCorrect()
```

---

## normalizeName

```
public void normalizeName()
```

Normalize the entity's name.

---

## toString

```
public java.lang.String toString()
```

---

(continued from last page)

## main

```
public static void main(java.lang.String[] a)
```

# tud.iir.knowledge

## Class Extractable

java.lang.Object

└-- tud.iir.knowledge.Extractable

All Implemented Interfaces:

java.io.Serializable

Direct Known Subclasses:

[Attribute](#), [Entity](#), [QA](#), [Snippet](#), [Event](#)

---

public abstract class **Extractable**

extends java.lang.Object

implements java.io.Serializable

The abstract class of what can be extracted.

**Author:**

David Urbansky

---

## Fields

### UNKNOWN

public static int **UNKNOWN**

---

### TRAINING

public static int **TRAINING**

---

### TESTING

public static int **TESTING**

---

## Constructors

### Extractable

public **Extractable**()

---

## Methods

### getID

public int **getID**()

---

## setID

```
public void setID(int id)
```

---

## getName

```
public java.lang.String getName()
```

---

## getSafeName

```
public java.lang.String getSafeName()
```

---

## setName

```
public void setName(java.lang.String name)
```

---

## getTrust

```
public double getTrust()
```

---

## setTrust

```
public void setTrust(double trust)
```

---

## getLastSearched

```
public java.util.Date getLastSearched()
```

---

## setLastSearched

```
public void setLastSearched(java.util.Date lastSearched)
```

---

## getExtractedAtAsUTCString

```
public java.lang.String getExtractedAtAsUTCString()
```

---

(continued from last page)

## getExtractedAt

```
public java.util.Date getExtractedAt()
```

---

## setExtractedAt

```
public void setExtractedAt(java.util.Date extractedAt)
```

---

## getSources

```
public Sources getSources()
```

---

## setSources

```
public void setSources(Sources sources)
```

---

## addSource

```
public void addSource(Source source)
```

---

## addSources

```
public void addSources(Sources sources)
```

---

## getType

```
public int getType()
```

---

## setType

```
public void setType(int type)
```

---

# tud.iir.knowledge

## Class Fact

```
java.lang.Object
└-- tud.iir.knowledge.Fact
```

---

```
public class Fact
extends java.lang.Object
```

The knowledge unit fact.

**Author:**  
David Urbansky

---

## Fields

### CORRECTNESS\_MARGIN

```
public static final double CORRECTNESS_MARGIN
```

some values cannot be accurate, allow a margin therefore  
Constant value: 0.15

## Constructors

### Fact

```
public Fact(Attribute attribute)
```

---

### Fact

```
public Fact(Attribute attribute,
FactValue value)
```

---

### Fact

```
public Fact(Attribute attribute,
            java.lang.String value,
            Source source,
            int extractionType)
```

---

## Methods

### getID

```
public java.lang.String getID()
```

Returns an identification string for the fact: "conceptAttribute".

---

(continued from last page)

**Returns:**

An identification string for the fact.

---

**getSamePowerFactValues**

```
public int getSamePowerFactValues(FactValue fv)
```

---

**getAttribute**

```
public Attribute getAttribute()
```

---

**setAttribute**

```
public void setAttribute(Attribute attribute)
```

---

**getValues**

```
public java.util.ArrayList getValues()
```

---

**getValues**

```
public java.util.ArrayList getValues(boolean sorted)
```

---

**getValues**

```
public java.util.ArrayList getValues(boolean sorted,  
                                     int limit)
```

---

**getFactValueForValue**

```
public FactValue getFactValueForValue(java.lang.String value)
```

---

**setValues**

```
public void setValues(java.util.ArrayList value)
```

---

**addFactValue**

```
public void addFactValue(FactValue factValue)
```



---

## getCorrectValue

```
public FactValue getCorrectValue()
```

For benchmarking set the correct value to compare with extracted ones.

**Returns:**  
The fact value.

---

## setCorrectValue

```
public void setCorrectValue(FactValue correctValue)
```

---

## getFactValue

```
public FactValue getFactValue()
```

Return fact value with highest corroboration.

**Returns:**  
The fact value.

---

## getValue

```
public java.lang.String getValue()
```

Returns the value of fact value with highest corroboration.

**Returns:**  
The value of fact value with highest corroboration.

---

## getCorroboration

```
public double getCorroboration()
```

Get corroboration for the value that is most likely.

**Returns:**  
The trust.

---

## isCorrect

```
public boolean isCorrect()
```

Returns true when given fact value is either correct or almost correct.

**Parameters:**  
`factValue` - The fact value.

**Returns:**  
True if it is set correct, else false.

---

## isCorrect

```
public boolean isCorrect(java.lang.String value)
```

---

---

## isAbsoluteCorrect

```
public boolean isAbsoluteCorrect()
```

Tell whether most likely fact value is correct.

### Returns:

True if the fact is absolutely correct.

---

## isAbsoluteCorrect

```
public boolean isAbsoluteCorrect(java.lang.String value)
```

---

## isAlmostCorrect

```
public boolean isAlmostCorrect()
```

---

## isAlmostCorrect

```
public boolean isAlmostCorrect(java.lang.String value)
```

---

## toString

```
public java.lang.String toString()
```

# tud.iir.knowledge

## Class FactValue

java.lang.Object

└--tud.iir.knowledge.FactValue

All Implemented Interfaces:

java.io.Serializable

---

```
public class FactValue
  extends java.lang.Object
  implements java.io.Serializable
```

The knowledge unit fact value.

**Author:**

David Urbansky

---

## Constructors

### FactValue

```
public FactValue(java.lang.String value,
                 Source source,
                 int extractionType)
```

## Methods

### getFact

```
public Fact getFact()
```

---

### setFact

```
public void setFact(Fact fact)
```

---

### getValue

```
public java.lang.String getValue()
```

---

### setValue

```
public void setValue(java.lang.String value)
```

---

## getOriginalValue

```
public java.lang.String getOriginalValue()
```

---

## setOriginalValue

```
public void setOriginalValue(java.lang.String originalValue)
```

---

## getSources

```
public java.util.ArrayList getSources()
```

---

## setSources

```
public void setSources(java.util.ArrayList sources)
```

---

## addSource

```
public void addSource(Source source)
```

---

## removeSource

```
public void removeSource(Source source)
```

---

## getExtractionTypes

```
public java.util.ArrayList getExtractionTypes(boolean oncePerSource)
```

Get extraction types used to extract that value.

### Parameters:

`oncePerSource` - If true each extraction type counts only once per source, e.g. 11 sentence extractions from two different pages will be 2 instead of 11.

### Returns:

An array of extraction types.

---

## getCorroboration

```
public double getCorroboration()
```

Get the corroboration for the fact value.

### Returns:

The trust.

---

## getCorroboration1

```
public double getCorroboration1()
```

Simple counting corroboration, the more sources the higher corroboration.

**Returns:**

The trust.

---

## getCorroboration2

```
public double getCorroboration2()
```

Variety corroboration. The more different extraction types and sources were used to extract that value, the higher the corroboration.

**Returns:**

The trust.

---

## getCorroboration3

```
public double getCorroboration3()
```

---

## getCorroboration4

```
public double getCorroboration4()
```

Extraction type trust and source applicability.

**Returns:**

The trust.

---

## getCorroboration5

```
public double getCorroboration5()
```

---

## getRelativeTrust

```
public double getRelativeTrust()
```

---

## getExtractedAt

```
public java.util.Date getExtractedAt()
```

---

## setExtractedAt

```
public void setExtractedAt(java.util.Date extractedAt)
```

---

---

## getTrust

```
public double getTrust()
```

---

## setTrust

```
public void setTrust(double trust)
```

---

## toString

```
public java.lang.String toString()
```

# tud.iir.knowledge

## Class HTMLSymbols

java.lang.Object

└--tud.iir.knowledge.HTMLSymbols

---

public class **HTMLSymbols**  
extends java.lang.Object

---

## Fields

### emptyWhitsp

public static final java.lang.String **emptyWhitsp**

---

### NBSP

public static final java.lang.String **NBSP**

Protected whitespace.

---

### NBSP2

public static final java.lang.String **NBSP2**

Protected whitespace.

---

### QUOT

public static final java.lang.String **QUOT**

Quotemark ".

---

### AMP

public static final java.lang.String **AMP**

Paragraph &.

---

### LT

public static final java.lang.String **LT**

Less then <.

---

### GT

public static final java.lang.String **GT**

Greater then >.

---

---

## AUML

```
public static final java.lang.String AUML
```

Letter Å.

---

## AAUML

```
public static final java.lang.String AAUML
```

Letter Ä.

---

## OUML

```
public static final java.lang.String OUML
```

Letter Ö.

---

## OOUML

```
public static final java.lang.String OOUML
```

Letter Ö.

---

## UUML

```
public static final java.lang.String UUML
```

Letter Ü.

---

## UUUML

```
public static final java.lang.String UUUML
```

Letter Ü.

---

## SZLIG

```
public static final java.lang.String SZLIG
```

Letter Š.

---

## NL

```
public static final java.lang.String NL
```

New Line \

---

## Constructors

### HTMLSymbols

```
public HTMLSymbols()
```

## Methods

---



(continued from last page)

## getHTMLSymboles

```
public static java.util.ArrayList getHTMLSymboles()
```

Returns all string arrays of HTML symbols.

A HTML symbol for example is    which stands for a whitespace.

A list-element consist of the HTML-code and the corresponding symbol.

# tud.iir.knowledge

## Class KeyWords

```
java.lang.Object
├-- tud.iir.knowledge.KeyWords
```

---

```
public final class KeyWords
extends java.lang.Object
```

---

### Fields

#### FIRST\_PRIORITY

```
public static final byte FIRST_PRIORITY
```

Constant value: 1

---

#### SECOND\_PRIORITY

```
public static final byte SECOND_PRIORITY
```

Constant value: 2

---

#### THIRD\_PRIORITY

```
public static final byte THIRD_PRIORITY
```

Constant value: 3

---

#### HTTP\_KEYWORDS

```
public static final java.lang.String HTTP_KEYWORDS
```

Keyowrds found in HTTP-header.

---

#### HEAD\_KEYWORDS

```
public static final java.lang.String HEAD_KEYWORDS
```

Keywords found in HTTP header of connections.

---

#### DATE\_BODY\_STRUC

```
public static final java.lang.String DATE_BODY_STRUC
```

Keywords found in HTML structure of documents.

---

(continued from last page)

## BODY\_CONTENT\_KEYWORDS

```
public static final java.lang.String BODY_CONTENT_KEYWORDS
```

Keywords found in HTML content of documents.

---

## firstPriorityKeywords

```
public static final java.lang.String firstPriorityKeywords
```

---

## secondPriorityKeywords

```
public static final java.lang.String secondPriorityKeywords
```

---

## thirdPriorityKexwords

```
public static final java.lang.String thirdPriorityKexwords
```

---

## allKeywords

```
public static final java.lang.String allKeywords
```

---

## Constructors

### KeyWords

```
public KeyWords()
```

# tud.iir.knowledge

## Class KnowledgeManager

java.lang.Object

└-- tud.iir.knowledge.KnowledgeManager

All Implemented Interfaces:

java.io.Serializable

---

```
public class KnowledgeManager
extends java.lang.Object
implements java.io.Serializable
```

TODO separate conceptual and instance knowledge (concept,attribute | entity,fact) The KnowledgeManager manages all other knowledge units.

**Author:**

David Urbansky

---

## Constructors

### KnowledgeManager

```
public KnowledgeManager()
```

## Methods

### serialize

```
public void serialize()
```

---

### addConcept

```
public void addConcept(Concept concept)
```

Add a concept to the KnowledgeManager, if a concept with the same name does not yet exist.

**Parameters:**

`concept` - The concept to add.

---

### addConcepts

```
public void addConcepts(java.util.Set concepts)
```

Add a set of concepts if they do not yet exist.

**Parameters:**

`concepts` - The set of concepts to add.

---

(continued from last page)

## mergeConcepts

```
public void mergeConcepts(java.util.HashSet concepts2)
```

In the extraction loop, the status is saved. The concepts in the saved status are not necessarily the updated ones from the database. We need to add all entities and the lastSearched field from the extraction status concepts to the loaded ones.

---

## getConcepts

```
public java.util.ArrayList getConcepts()
```

---

## getConcepts

```
public java.util.ArrayList getConcepts(boolean sortedByDate)
```

---

## getConcept

```
public Concept getConcept(java.lang.String conceptName)
```

Get a certain concept by name.

**Parameters:**

`conceptName` - The name of the concept.

**Returns:**

The concept.

---

## getConcept

```
public Concept getConcept(java.lang.String conceptName,  
                           boolean useSynonyms)
```

---

## getConcept

```
public Concept getConcept(int conceptId)
```

Get a certain concept by id.

**Parameters:**

`conceptId` - The id of the concept.

**Returns:**

The concept.

---

## removeConcept

```
public void removeConcept(java.lang.String conceptName)
```

---

(continued from last page)

## removeConcept

```
public void removeConcept(Concept concept)
```

---

## createSnippetBenchmarks

```
public void createSnippetBenchmarks()
```

---

## createBenchmarkConcepts

```
public void createBenchmarkConcepts()
```

---

## createBenchmarkConcepts

```
public void createBenchmarkConcepts(boolean imageAttributes)
```

---

## setCorrectValues

```
public void setCorrectValues()
```

Set the correct values for the benchmark concepts, entities and attributes.

---

## evaluateBenchmarkExtractions

```
public void evaluateBenchmarkExtractions()
```

---

## evaluateBenchmarkExtractionsGetPAR

```
public java.lang.Double[] evaluateBenchmarkExtractionsGetPAR()
```

---

## fillDomainsForFactExtractionTest

```
public void fillDomainsForFactExtractionTest()
```

---

## updateTrust

```
public boolean updateTrust()
```

---

## updateTrust

```
public boolean updateTrust(boolean saveLogs)
```

---

(continued from last page)

---

## saveExtractions

```
public void saveExtractions()
```

---

## calculateAttributeSynonyms

```
public void calculateAttributeSynonyms()
```

Try to connect attributes that might be synonyms. Do not consider manually defined attributes as pairs. Calculate a trust for each attribute pair of a concept and connect top pairs with trust above certain threshold.

---

## main

```
public static void main(java.lang.String[] a)
```

# tud.iir.knowledge

## Class QA

java.lang.Object

└- [tud.iir.knowledge.Extractable](#)

└- tud.iir.knowledge.QA

All Implemented Interfaces:  
java.io.Serializable

public class QA  
extends [Extractable](#)

## Constructors

### QA

public QA([QASite](#) qaSite)

## Methods

### getQuestion

public java.lang.String **getQuestion**()

### setQuestion

public void **setQuestion**(java.lang.String question,  
java.lang.String url,  
java.lang.String xPath)

### getAnswers

public java.util.ArrayList **getAnswers**()

### addAnswer

public boolean **addAnswer**(java.lang.String answer,  
java.lang.String url,  
java.lang.String xPath)



(continued from last page)

## toString

```
public java.lang.String toString()
```

```
+tud.iir.knowledge.RegExp
```

David Urbansky, Martin Gregor

---

## DATE4

```
public static final java.lang.String DATE4
```

Constant value: `(\w){3,9}\s(\d){1,2}(th)?((\,)|(\s))+([''])?(\d){2,4}`

---

## DATE\_ISO8601\_YMD\_T

```
public static final java.lang.String DATE_ISO8601_YMD_T
```

[ISO8601](#) YYYY-MM-DD TIME+UTC.

---

## DATE\_ISO8601\_YMD\_SEPARATOR\_T

```
public static final java.lang.String DATE_ISO8601_YMD_SEPARATOR_T
```

---

## DATE\_ISO8601\_YMD

```
public static final java.lang.String DATE_ISO8601_YMD
```

ISO8601 YYYY-MM-DD .

---

## DATE\_ISO8601\_YMD\_SEPARATOR

```
public static final java.lang.String DATE_ISO8601_YMD_SEPARATOR
```

ISO8601 YYYY-MM-DD .

---

## DATE\_ISO8601\_YM

```
public static final java.lang.String DATE_ISO8601_YM
```

ISO8601 YYYY-MM .

---

## DATE\_ISO8601\_YWD\_T

```
public static final java.lang.String DATE_ISO8601_YWD_T
```

ISO8601 YYYY-WW-D TIME+UTC .

---

## DATE\_ISO8601\_YWD

```
public static final java.lang.String DATE_ISO8601_YWD
```

ISO8601 YYYY-WW-D .

---

## DATE\_ISO8601\_YW

```
public static final java.lang.String DATE_ISO8601_YW
```

ISO8601 YYYY-WW .

---

(continued from last page)

---

## DATE\_ISO8601\_YD\_T

```
public static final java.lang.String DATE_ISO8601_YD_T
```

ISO8601 YYYY-DDD TIME+UTC.

---

## DATE\_ISO8601\_YD

```
public static final java.lang.String DATE_ISO8601_YD
```

ISO8601 YYYY-DDD .

---

## DATE\_ISO8601\_YMD\_NO

```
public static final java.lang.String DATE_ISO8601_YMD_NO
```

Year, month and day written without separator.  
YYYYMMDD

---

## DATE\_ISO8601\_YWD\_NO

```
public static final java.lang.String DATE_ISO8601_YWD_NO
```

Year, month and day written without separator.  
YYYYWWD

---

## DATE\_ISO8601\_YW\_NO

```
public static final java.lang.String DATE_ISO8601_YW_NO
```

Year and month written without separator.  
YYYYWW

---

## DATE\_ISO8601\_YD\_NO

```
public static final java.lang.String DATE_ISO8601_YD_NO
```

Year and month written without separator.  
YYYYDDD

---

## DATE\_URL\_D

```
public static final java.lang.String DATE_URL_D
```

Dates in URL. YYYY\_MM\_DD .  
"\_" can also be "." or "-" or "/"

---

## DATE\_URL\_MMMM\_D

```
public static final java.lang.String DATE_URL_MMMM_D
```

Dates in URL. YYYY\_MM\_DD .  
"\_" can also be "." or "-" or "/"

---

## DATE\_URL

```
public static final java.lang.String DATE_URL
```

---

(continued from last page)

Dates in URL. YYYY\_MM .  
"\_" can also be "." or "-" or "/"

---

## DATE\_URL\_SPLIT

```
public static final java.lang.String DATE_URL_SPLIT
```

Date in URL, that can be split by folders between year an month.  
YYYY\...\MM\DD

---

## DATE\_EU\_D\_MM\_Y

```
public static final java.lang.String DATE_EU_D_MM_Y
```

European date. DD.MM.YYYY .

---

## DATE\_EU\_D\_MM\_Y\_T

```
public static final java.lang.String DATE_EU_D_MM_Y_T
```

European date. DD.MM.YYYY HH:MM:SS+UTC.

---

## DATE\_EU\_MM\_Y

```
public static final java.lang.String DATE_EU_MM_Y
```

European date. MM.YYYY .

---

## DATE\_EU\_D\_MM

```
public static final java.lang.String DATE_EU_D_MM
```

European date. DD.MM. .

---

## DATE\_EU\_D\_MMMM\_Y

```
public static final java.lang.String DATE_EU_D_MMMM_Y
```

European date. DD. MMMM YYYY .

---

## DATE\_EU\_D\_MMMM

```
public static final java.lang.String DATE_EU_D_MMMM
```

European date. DD.MMMM .

---

## DATE\_EU\_D\_MMMM\_Y\_T

```
public static final java.lang.String DATE_EU_D_MMMM_Y_T
```

European date. DD. MMMM YYYY HH:MM:SS +UTC .

---

## DATE\_USA\_MM\_D\_Y

```
public static final java.lang.String DATE_USA_MM_D_Y
```

American date. MM/DD/YYYY.

---

## DATE\_USA\_MM\_D\_Y\_T

```
public static final java.lang.String DATE_USA_MM_D_Y_T
```

American date MM/DD/YYYY. HH:MM:SS +UTC.

---

## DATE\_USA\_MM\_D\_Y\_SEPARATOR\_1

```
public static final java.lang.String DATE_USA_MM_D_Y_SEPARATOR_1
```

Constant value: `((([0-2])|(0?[1-9]))\\.((((0[1-9])|([12][0-9])|(3[01])))|((([1-9])|([12][0-9])|(3[01]))))\\((((\\d){4})|(('?)(\\d){2})))`

---

## DATE\_USA\_MM\_D\_Y\_SEPARATOR\_2

```
public static final java.lang.String DATE_USA_MM_D_Y_SEPARATOR_2
```

Constant value: `((([0-2])|(0?[1-9]))-((((0[1-9])|([12][0-9])|(3[01])))|((([1-9])|([12][0-9])|(3[01]))))-((\\d){4})|(('?)(\\d){2})))`

---

## DATE\_USA\_MM\_D\_Y\_SEPARATOR\_3

```
public static final java.lang.String DATE_USA_MM_D_Y_SEPARATOR_3
```

Constant value: `((([0-2])|(0?[1-9]))_((((0[1-9])|([12][0-9])|(3[01])))|((([1-9])|([12][0-9])|(3[01]))))_((\\d){4})|(('?)(\\d){2})))`

---

## DATE\_USA\_MM\_D\_Y\_SEPARATOR

```
public static final java.lang.String DATE_USA_MM_D_Y_SEPARATOR
```

American date. MM/DD/YYYY.

---

## DATE\_USA\_MM\_Y

```
public static final java.lang.String DATE_USA_MM_Y
```

American date. MM/YYYY .

---

## DATE\_USA\_MM\_D

```
public static final java.lang.String DATE_USA_MM_D
```

American date. MM/DD .

---

## DATE\_USA\_MMMM\_D\_Y

```
public static final java.lang.String DATE_USA_MMMM_D_Y
```

American date. MMMM DD(st), YYYY .

---

## DATE\_USA\_MMMM\_D\_Y\_T

```
public static final java.lang.String DATE_USA_MMMM_D_Y_T
```

American date. MMMM DD(st), YYYY HH:MM:SS +UTC.

---

---

## DATE\_USA\_MMMM\_D

```
public static final java.lang.String DATE_USA_MMMM_D
```

American date. MMMM DD(st) .

---

## DATE\_EUSA\_MMMM\_Y

```
public static final java.lang.String DATE_EUSA_MMMM_Y
```

American and European date. "MMMM YYYY" .

---

## DATE\_RFC\_1123

```
public static final java.lang.String DATE_RFC_1123
```

RFC 1123. WD, DD MMM YYYY HH:MM:SS TZ .

---

## DATE\_RFC\_1036

```
public static final java.lang.String DATE_RFC_1036
```

RFC 1036. WWD, DD-MMM-YYYY HH:MM:SS TZ .

---

## DATE\_RFC\_1123\_UTC

```
public static final java.lang.String DATE_RFC_1123_UTC
```

RFC 1123. WD, DD MMM YYYY HH:MM:SS +UTC .

---

## DATE\_RFC\_1036\_UTC

```
public static final java.lang.String DATE_RFC_1036_UTC
```

RFC 1036. WWD, DD-MMM-YYYY HH:MM:SS +UTC .

---

## DATE\_ANSI\_C

```
public static final java.lang.String DATE_ANSI_C
```

ANSI C's ascitime. WD MMM DD\_1 HH:MM:SS YYYY .

---

## DATE\_ANSI\_C\_TZ

```
public static final java.lang.String DATE_ANSI_C_TZ
```

ANSI C's ascitime with time difference to UTC. WD MMM DD\_1 HH:MM:SS YYYY +UTC.

---

## COLON\_FACT\_REPRESENTATION

```
public static final java.lang.String COLON_FACT_REPRESENTATION
```

Constant value: `[A-Za-z0-9/() ]{1,20}:\s?((([A-Z]+|[a-z]+|[0-9.]+[A-Z]{1,2}(\s|,$)|[0-9.]+[a-z]{1,4}|[0-9.]+)))+((\s|,)+([A-Z]+|[a-z]+|[0-9.]+[A-Z]{1,2}(\s|,$)|[0-9.]+[a-z]{1,4}|[0-9.]+))*`

---

(continued from last page)

# Constructors

## RegExp

```
public RegExp()
```

# Methods

## getRegExp

```
public static java.lang.String getRegExp(int valueType)
```

---

## getAllRegExp

```
public static java.lang.Object[] getAllRegExp()
```

Get all regular Expressions.

Returns:

---

## getRFCRegExp

```
public static java.lang.Object[] getRFCRegExp()
```

---

## getIncTimeRegExp

```
public static java.lang.Object[] getIncTimeRegExp()
```

---

## get3PartRegExp

```
public static java.lang.Object[] get3PartRegExp()
```

---

## get2PartRegExp

```
public static java.lang.Object[] get2PartRegExp()
```

---

## get1PartRegExp

```
public static java.lang.Object[] get1PartRegExp()
```

---



(continued from last page)

## getOthersRegExp

```
public static java.lang.Object[] getOthersRegExp()
```

---

## getURLRegExp

```
public static java.lang.Object[] getURLRegExp()
```

# For URL-dates.

Get an ordered array of [regular expressions](#) to match the longest possible string.

We need order because short regular expression matches also longer ones.  
E.g.: So we get for 2010-07-20 a match for YYYY-MM and YYYY-MM-DD. But last one would be more specific.

**Returns:**

Array with [regular expressions](#)

---

## getHTTPRegExp

```
public static java.lang.Object[] getHTTPRegExp()
```

# For HTTP-Header-dates.

Get an ordered array of [regular expressions](#) to match the longest possible string.

We need order because short regular expression matches also longer ones.  
E.g.: So we get for 2010-07-20 a match for YYYY-MM and YYYY-MM-DD. But last one would be more specific.

**Returns:**

Array with [regular expressions](#)

---

## getHEADRegExp

```
public static java.lang.Object[] getHEADRegExp()
```

# For HTML-head-dates..

Get an ordered array of [regular expressions](#) to match the longest possible string.

We need order because short regular expression matches also longer ones.  
E.g.: So we get for 2010-07-20 a match for YYYY-MM and YYYY-MM-DD. But last one would be more specific.

**Returns:**

Array with [regular expressions](#)

---

## getTimezones

```
public static java.lang.String getTimezones()
```

# tud.iir.knowledge Class Snippet

```
java.lang.Object
├── tud.iir.knowledge.Extractable
│   └── tud.iir.knowledge.Snippet
```

All Implemented Interfaces:  
java.io.Serializable

```
public class Snippet
extends Extractable
```

The knowledge unit snippet contains the snippet text, a reference to the entity it belongs to, a reference to the aggregated result it was extracted from and a feature vector containing features about the snippet which might be used for regression learning.

**Author:**  
Christopher Friedrich

## Constructors

### Snippet

```
public Snippet(Entity entity,
               AggregatedResult webresult,
               java.lang.String text)
```

## Methods

### getFeature

```
public double getFeature(java.lang.String name)
```

### setFeature

```
public void setFeature(java.lang.String name,
                      double value)
```

### getFeatures

```
public java.util.Map getFeatures()
```

### getEntity

```
public Entity getEntity()
```

---

## getAggregatedResult

```
public AggregatedResult getAggregatedResult()
```

---

## getText

```
public java.lang.String getText()
```

---

## startsWithEntity

```
public boolean startsWithEntity()
```

Whether the snippet starts with a mentioning of the related entity.

**Returns:**

True, if it starts with an entity and False otherwise.

---

## classify

```
public double classify()
```

Calculate the regression value using the SnippetClassifier on a trained model. XXX what's going on here?

**Returns:**

Regression value.

---

## getRegressionRank

```
public double getRegressionRank()
```

**Deprecated.** *Alias for classify().*

---

## toString

```
public final java.lang.String toString()
```

# tud.iir.knowledge

## Class Source

java.lang.Object

└─ tud.iir.knowledge.Source

All Implemented Interfaces:

java.io.Serializable

---

```
public class Source
extends java.lang.Object
implements java.io.Serializable
```

A source from which an extraction was performed.

**Author:**

David Urbansky, Christopher Friedrich

---

## Constructors

### Source

```
public Source(java.lang.String url,
              double trust,
              int extractionType)
```

---

### Source

```
public Source(java.lang.String url,
              int extractionType)
```

---

### Source

```
public Source(java.lang.String url,
              double trust)
```

---

### Source

```
public Source(java.lang.String url)
```

---

## Methods

### getFactValue

```
public FactValue getFactValue()
```

---

## setFactValue

```
public void setFactValue(FactValue factValue)
```

---

---

## getUrl

```
public java.lang.String getUrl()
```

---

---

## setUrl

```
public void setUrl(java.lang.String url)
```

---

---

## getTrust

```
public double getTrust()
```

---

---

## setTrust

```
public void setTrust(double trust)
```

---

---

## getExtractionType

```
public int getExtractionType()
```

Count the number of same fact values that have been extracted from this source for the fact. The more same values the more trust for one certain value.

**Returns:**

The number of same values.

---

---

## setExtractionType

```
public void setExtractionType(int extractionType)
```

---

---

## getID

```
public int getID()
```

---

---

## setID

```
public void setID(int id)
```

---

---

## getPageRank

```
public double getPageRank()
```

Get Google's PageRank for the source URL.

**Returns:**

Google Page Rank for source URL.

---

## getMainContent

```
public java.lang.String getMainContent()
```

Get the main content block from the source URL page.

**Returns:**

The main content string.

---

## setMainContent

```
public void setMainContent(java.lang.String mainContent)
```

Override the main content block for this object.

---

## getTLD

```
public java.lang.String getTLD()
```

Get the top level domain (TLD) of the source URL.

**Returns:**

The TLD of the source URL.

---

## equals

```
public boolean equals(java.lang.Object obj)
```

---

## toString

```
public java.lang.String toString()
```

---

# tud.iir.knowledge

## Class Sources

```

java.lang.Object
  |-- java.util.AbstractCollection
        |-- java.util.AbstractSet
              |-- java.util.HashSet
                    |-- tud.iir.knowledge.Sources

```

### All Implemented Interfaces:

java.io.Serializable, java.util.Collection, java.util.Set, java.io.Serializable, java.lang.Cloneable, java.util.Set

```

public class Sources
extends java.util.HashSet
implements java.util.Set, java.lang.Cloneable, java.io.Serializable, java.util.Set,
java.util.Collection, java.io.Serializable

```

A set of sources.

### Parameters:

S

### Author:

David Urbansky

## Constructors

### Sources

```
public Sources()
```

## Methods

### contains

```
public final boolean contains(java.lang.Object o)
```



---

Package  
**tud.iir.multimedia**

# tud.iir.multimedia

## Class ExtractedImage

```
java.lang.Object
├── tud.iir.multimedia.Image
│   └── tud.iir.multimedia.ExtractedImage
```

---

```
public class ExtractedImage
    extends Image
```

An extracted image.  
**Author:**  
David Urbansky

---

## Constructors

### ExtractedImage

```
public ExtractedImage()
```

## Methods

### getRankCount

```
public int getRankCount()
```

---

### setRankCount

```
public void setRankCount(int rankCount)
```

---

### addRanking

```
public void addRanking(int ranking)
```

---

### getDuplicateCount

```
public int getDuplicateCount()
```

---

### setDuplicateCount

```
public void setDuplicateCount(int duplicateCount)
```

---

---

## addDuplicate

```
public void addDuplicate()
```

---

## getRanking

```
public double getRanking()
```

---

## toString

```
public java.lang.String toString()
```

# tud.iir.multimedia

## Class ExtractedImageComparator

java.lang.Object

└─ tud.iir.multimedia.ExtractedImageComparator

### All Implemented Interfaces:

java.io.Serializable, java.util.Comparator

---

```
public class ExtractedImageComparator
    extends java.lang.Object
    implements java.util.Comparator, java.io.Serializable
```

Sort extracted images.

### Author:

David Urbansky

---

## Constructors

### ExtractedImageComparator

```
public ExtractedImageComparator()
```

## Methods

### compare

```
public int compare(ExtractedImage image1,
                  ExtractedImage image2)
```

Higher ranking first.

### Parameters:

image1 - Image1

image2 - Image2

# tud.iir.multimedia

## Class Image

java.lang.Object

└─ tud.iir.multimedia.Image

Direct Known Subclasses:

[ExtractedImage](#)

---

```
public class Image
extends java.lang.Object
```

An image.

**Author:**

David Urbansky

---

## Constructors

### Image

```
public Image()
```

## Methods

### getURL

```
public java.lang.String getURL()
```

---

### setURL

```
public void setURL(java.lang.String url)
```

---

### getWidth

```
public int getWidth()
```

---

### setWidth

```
public void setWidth(int width)
```

---

### getHeight

```
public int getHeight()
```

---

## setHeight

```
public void setHeight(int height)
```

---

## getWidthHeightRatio

```
public double getWidthHeightRatio()
```

---

## getImageContent

```
public java.awt.image.BufferedImage getImageContent()
```

---

## setImageContent

```
public void setImageContent(java.awt.image.BufferedImage imageContent)
```

# tud.iir.multimedia

## Class ImageHandler

java.lang.Object

└─ tud.iir.multimedia.ImageHandler

---

public class **ImageHandler**  
extends java.lang.Object

A handler for images.

**Author:**

David Urbansky

---

## Fields

### MSE

public static final int **MSE**

Constant value: 1

---

### MINKOWSKI

public static final int **MINKOWSKI**

Constant value: 2

---

### DIFFG

public static final int **DIFFG**

Constant value: 3

---

## Constructors

### ImageHandler

public **ImageHandler**()

---

## Methods

### load

public static java.awt.image.BufferedImage **load**(java.lang.String url)

---

(continued from last page)

---

## getMatchingImageURL

```
public static java.lang.String getMatchingImageURL(java.util.ArrayList images)
```

---

## getMatchingImageURLs

```
public static java.lang.String[] getMatchingImageURLs(java.util.ArrayList images,  
int matchingNumber)
```

---

## rescaleImage

```
public static java.awt.image.BufferedImage rescaleImage(java.lang.String imageURL,  
int width)
```

---

## rescaleImageOptimal

```
public static java.awt.image.BufferedImage  
rescaleImageOptimal(java.awt.image.BufferedImage bufferedImage,  
int newWidth,  
boolean fit)
```

Rescaling an image using JAI SubsampleAverage for downscaling and getScaledInstance for upscaling. This produces smooth images but upscaling is slightly slower.

### Parameters:

`bufferedImage` - The input image.

`newWidth` - The desired new width (size) of the image.

`fit` - If true, the newWidth will be the maximum side length of the image. Default is false.

### Returns:

The scaled image.

---

## rescaleImage

```
public static java.awt.image.BufferedImage rescaleImage(java.awt.image.BufferedImage  
bufferedImage,  
int newWidth,  
boolean fit)
```

Rescaling an image using JAI SubsampleAverage. The image looks smooth after rescaling.

### Parameters:

`bufferedImage` - The input image.

`newWidth` - The desired new width (size) of the image.

`fit` - If true, the newWidth will be the maximum side length of the image. Default is false.

### Returns:

The scaled image.

---

## rescaleImage

```
public static java.awt.image.BufferedImage rescaleImage(java.awt.image.BufferedImage  
bufferedImage,  
int newWidth)
```

---



---

## rescaleImage2

```
public static java.awt.image.BufferedImage rescaleImage2( java.awt.image.BufferedImage
bufferedImage,
    int newWidth,
    boolean fit)
```

Rescaling an image using JAI Scale descriptor. The image does not look smooth after rescaling.

### Parameters:

`bufferedImage` - The input image.

`newWidth` - The desired new width (size) of the image.

`fit` - If true, the `newWidth` will be the maximum side length of the image. Default is false.

### Returns:

The scaled image.

---

## rescaleImage2

```
public static java.awt.image.BufferedImage rescaleImage2( java.awt.image.BufferedImage
bufferedImage,
    int newWidth)
```

---

## rescaleImage3

```
public static java.awt.image.BufferedImage rescaleImage3( java.awt.image.BufferedImage
bufferedImage,
    int newWidth,
    boolean fit)
```

Rescaling an image using `java.awt.Image.getScaledInstance`. The image looks smooth after rescaling.

### Parameters:

`bufferedImage` - The input image.

`newWidth` - The desired new width (size) of the image.

`fit` - If true, the `newWidth` will be the maximum side length of the image. Default is false.

### Returns:

The scaled image.

---

## rescaleImage3

```
public static java.awt.image.BufferedImage rescaleImage3( java.awt.image.BufferedImage
bufferedImage,
    int newWidth)
```

---

## rescaleImage\_broken

```
public static java.awt.image.BufferedImage
rescaleImage_broken( java.awt.image.BufferedImage bufferedImage,
    int width)
```

**Deprecated.**

---

(continued from last page)

**Parameters:**

bufferedImage  
width

**Returns:**

---

## downloadAndSave

```
public static void downloadAndSave(java.lang.String url,  
    java.lang.String savePath)
```

---

## getAverageGray

```
public static float getAverageGray(java.awt.image.BufferedImage bufferedImage)
```

---

## getSimilarity

```
public static double getSimilarity(java.awt.image.BufferedImage image1,  
    java.awt.image.BufferedImage image2,  
    int measure)
```

---

## getMeanSquareError

```
public static double getMeanSquareError(java.awt.image.BufferedImage image1,  
    java.awt.image.BufferedImage image2)
```

---

## getMinkowskiSimilarity

```
public static double getMinkowskiSimilarity(java.awt.image.BufferedImage image1,  
    java.awt.image.BufferedImage image2)
```

---

## getGrayDifference

```
public static double getGrayDifference(java.awt.image.BufferedImage image1,  
    java.awt.image.BufferedImage image2)
```

---

## toGrayScale

```
public static java.awt.image.BufferedImage toGrayScale(java.awt.image.BufferedImage  
bufferedImage)
```

## isDuplicate

```
public static boolean isDuplicate(java.awt.image.BufferedImage image1,  
    java.awt.image.BufferedImage image2)
```

---

## saveImage

```
public static boolean saveImage(java.awt.image.BufferedImage image,  
    java.lang.String fileType,  
    java.lang.String filePath)
```

Save an image to disk. This methods wraps the ImageIO.write method and does error handling.

### Parameters:

image - The image to save.  
fileType - The image type (e.g. "jpg")  
filePath - The path where the image should be saved.

### Returns:

True if the image was saved successfully, false otherwise.

---

## saveImage

```
public static boolean saveImage(java.awt.image.BufferedImage image,  
    java.lang.String fileType,  
    java.lang.String filePath,  
    float quality)
```

---

## saveImage2

```
public static boolean saveImage2(java.awt.image.BufferedImage image,  
    java.lang.String fileType,  
    java.lang.String filePath)
```

Save an image to disk. This methods wraps the ImageIO.write method and does error handling.

### Parameters:

image - The image to save.  
fileType - The image type (e.g. "jpg")  
filePath - The path where the image should be saved.

### Returns:

True if the image was saved successfully, false otherwise.

---

## saveImage3

```
public static boolean saveImage3(java.awt.image.BufferedImage bufferedImage,  
    java.lang.String fileType,  
    java.lang.String filePath)
```

Save an image to disk. This methods wraps the JAI.create method and does error handling.

### Parameters:

image - The image to save.  
fileType - The image type (e.g. "jpg")  
filePath - The path where the image should be saved.

(continued from last page)

**Returns:**

True if the image was saved successfully, false otherwise.

---

**main**

```
public static void main(java.lang.String[] args)
```

---

Package  
**tud.iir.news**

# tud.iir.news

## Class CheckApproach

```
java.lang.Object
├-- java.lang.Enum
│   └-- tud.iir.news.CheckApproach
```

### All Implemented Interfaces:

java.io.Serializable, java.lang.Comparable

```
public final class CheckApproach
extends java.lang.Enum
```

Approach used for setting the interval a feed is checked for updates.

### See Also:

[FeedChecker](#)

### Author:

Klemens Muthmann

## Fields

### CHECK\_FIXED

```
public static final tud.iir.news.CheckApproach CHECK_FIXED
```

Check each feed at a fixed interval.

### CHECK\_ADAPTIVE

```
public static final tud.iir.news.CheckApproach CHECK_ADAPTIVE
```

Check each feed and learn its update times.

### CHECK\_PROBABILISTIC

```
public static final tud.iir.news.CheckApproach CHECK_PROBABILISTIC
```

Check each feed and adapt to its update rate.

## Methods

### values

```
public static CheckApproach\[\] values()
```

### valueOf

```
public static CheckApproach valueOf(java.lang.String name)
```

# tud.iir.news

## Class DatasetCreator

java.lang.Object

└─ tud.iir.news.DatasetCreator

---

public class **DatasetCreator**  
extends java.lang.Object

Creates a dataset of feeds.

**Since:**

1.0

**Author:**

klemens.muthmann@googlemail.com

**Version:**

1.0

---

## Constructors

### DatasetCreator

public **DatasetCreator**()

## Methods

### main

public static void **main**(java.lang.String[] args)

Run creation of the feed dataset from all feeds in the database if possible.

**Parameters:**

args

# tud.iir.news

## Class Feed

```
java.lang.Object
└─ tud.iir.news.Feed
```

---

```
public class Feed
extends java.lang.Object
```

Represents a news feed.

**Author:**

Philipp Katz, David Urbansky, klemens.muthmann@googlemail.com

---

## Fields

### FORMAT\_ATOM

```
public static final int FORMAT_ATOM
```

different formats of feeds; this has just informational character; the parser of the aggregator will determine the feed's format automatically.  
Constant value: 1

---

### FORMAT\_RSS

```
public static final int FORMAT_RSS
```

Constant value: 2

---

### TEXT\_TYPE\_UNDETERMINED

```
public static final int TEXT_TYPE_UNDETERMINED
```

Constant value: 0

---

### TEXT\_TYPE\_NONE

```
public static final int TEXT_TYPE_NONE
```

Constant value: 1

---

### TEXT\_TYPE\_PARTIAL

```
public static final int TEXT_TYPE_PARTIAL
```

Constant value: 2

---

### TEXT\_TYPE\_FULL

```
public static final int TEXT_TYPE_FULL
```

---



(continued from last page)

Constant value: 3

## Constructors

### Feed

```
public Feed()
```

### Feed

```
public Feed(java.lang.String feedUrl)
```

## Methods

### getId

```
public int getId()
```

### setId

```
public void setId(int id)
```

### getFeedUrl

```
public java.lang.String getFeedUrl()
```

### setFeedUrl

```
public void setFeedUrl(java.lang.String feedUrl)
```

### getSiteUrl

```
public java.lang.String getSiteUrl()
```

### setSiteUrl

```
public void setSiteUrl(java.lang.String pageUrl)
```

(continued from last page)

## getTitle

```
public java.lang.String getTitle()
```

---

## setTitle

```
public void setTitle(java.lang.String title)
```

---

## getFormat

```
public int getFormat()
```

---

## setFormat

```
public void setFormat(int format)
```

---

## getAdded

```
public java.util.Date getAdded()
```

---

## getAddedSQLTimestamp

```
public java.sql.Timestamp getAddedSQLTimestamp()
```

---

## setAdded

```
public void setAdded(java.util.Date added)
```

---

## getLanguage

```
public java.lang.String getLanguage()
```

---

## setLanguage

```
public void setLanguage(java.lang.String language)
```

---

## getTextType

```
public int getTextType()
```

---

(continued from last page)

---

## setTextType

```
public void setTextType(int textType)
```

---

## setEntries

```
public void setEntries(java.util.List entries)
```

---

## getEntries

```
public java.util.List getEntries()
```

---

## setChecks

```
public void setChecks(int checks)
```

---

## increaseChecks

```
public void increaseChecks()
```

---

## getChecks

```
public int getChecks()
```

---

## setMaxCheckInterval

```
public void setMaxCheckInterval(int maxCheckInterval)
```

---

## getMaxCheckInterval

```
public int getMaxCheckInterval()
```

---

## setMinCheckInterval

```
public void setMinCheckInterval(int minCheckInterval)
```

---

(continued from last page)

---

## getMinCheckInterval

```
public int getMinCheckInterval()
```

---

## setLastHeadlines

```
public void setLastHeadlines(java.lang.String lastHeadlines)
```

---

## getLastHeadlines

```
public java.lang.String getLastHeadlines()
```

---

## setUnreachableCount

```
public void setUnreachableCount(int unreachableCount)
```

---

## getUnreachableCount

```
public int getUnreachableCount()
```

---

## setLastFeedEntry

```
public void setLastFeedEntry(java.util.Date lastFeedEntry)
```

---

## getLastFeedEntry

```
public java.util.Date getLastFeedEntry()
```

---

## getLastFeedEntrySQLTimestamp

```
public java.sql.Timestamp getLastFeedEntrySQLTimestamp()
```

---

## setMeticulousPostDistribution

```
public void setMeticulousPostDistribution(java.util.Map meticulousPostDistribution)
```

---

## getMeticulousPostDistribution

```
public java.util.Map getMeticulousPostDistribution()
```

---

---

## oneFullDayHasBeenSeen

```
public boolean oneFullDayHasBeenSeen()
```

Check whether the checked entries in the feed were spread over at least one day yet. That means in every minute of the day the chances field should be greater or equal to one.

**Returns:**

True, if the entries span at least one day, false otherwise.

---

## setUpdateClass

```
public void setUpdateClass(int updateClass)
```

---

## getUpdateClass

```
public int getUpdateClass()
```

Returns the update class of the feed which is one of the following: [FeedClassifier.CLASS\\_CONSTANT](#), [FeedClassifier.CLASS\\_CHUNKED](#), [FeedClassifier.CLASS\\_SLICED](#), [FeedClassifier.CLASS\\_ZOMBIE](#), [FeedClassifier.CLASS\\_UNKNOWN](#) or [FeedClassifier.CLASS\\_ON\\_THE\\_FLY](#)

**Returns:**

The classID of the class. You can get the name using `getClassName()`

---

## toString

```
public java.lang.String toString()
```

---

## getLastChecked

```
public final java.util.Date getLastChecked()
```

**Returns:**

The date this feed was checked for updates the last time.

---

## setLastChecked

```
public final void setLastChecked(java.util.Date lastChecked)
```

**Parameters:**

`lastChecked` - The date this feed was checked for updates the last time.

---

# tud.iir.news

## Class FeedChecker

java.lang.Object

└─ tud.iir.news.FeedChecker

```
public final class FeedChecker
extends java.lang.Object
```

The FeedChecker reads news from feeds in a database. It learns when it is necessary to check the feed again for news.

**Author:**

David Urbansky

## Fields

### LOGGER

```
public static final Logger LOGGER
```

the logger for this class

## Methods

### getInstance

```
public static FeedChecker getInstance()
```

The FeedReader is singleton, get the instance here.

**Returns:**

The FeedReader instance.

### startContinuousReading

```
public void startContinuousReading(int duration)
```

Continuously read feeds.

**Parameters:**

`duration` - Time in milliseconds after it should stop reading, -1 means no time limit.

### startContinuousReading

```
public void startContinuousReading()
```

Start continuous reading without a time limit.

### stopContinuousReading

```
public void stopContinuousReading()
```

Stop all timers, no reading will be performed after stopping the reader.

---

## updateCheckIntervals

```
public void updateCheckIntervals(Feed feed)
```

Update the check interval depending on the chosen approach. Update the feed accordingly and return it. TODO this method is insanely long, break it down!

### Parameters:

`feed` - The feed to update.

`entries` - A list of entries of that feed. They are given in order to save the time here to retrieve them first.

### Returns:

The updated feed.

---

## setFeedProcessingAction

```
public void setFeedProcessingAction(FeedProcessingAction feedProcessingAction)
```

---

## getFeedProcessingAction

```
public FeedProcessingAction getFeedProcessingAction()
```

---

## setCheckApproach

```
public void setCheckApproach(CheckApproach checkApproach,  
    boolean resetLearnedValues)
```

Set the approach for checking feeds for news. Once an approach is chosen it cannot be changed (meta information is saved in the feed store) unless you reset the learned data.

### Parameters:

`checkApproach` - The updating approach, can be one of `CHECK_FIXED`, `CHECK_ADAPTIVE`, or `CHECK_PROBABILISTIC`

`resetLearnedValues` - If true, learned and calculated values such as check intervals etc. are reset and are retrained using the new check approach.

---

## getCheckApproach

```
public CheckApproach getCheckApproach()
```

---

## getCheckApproachName

```
public java.lang.String getCheckApproachName()
```

Get the human readable name of the chosen check approach.

---

## setCheckInterval

```
public void setCheckInterval(int checkInterval)
```

Set a fixed check interval in minutes. This is only effective if the `checkType` is set to `CHECK_FIXED`.

---

(continued from last page)

**Parameters:**`checkInterval` - Fixed check interval in minutes.

---

**getCheckInterval**

```
public int getCheckInterval()
```

---

**main**

```
public static void main(java.lang.String[] args)
```

Sample usage. Command line: parameters: checkType("cf" or "ca" or "cp") runtime(in minutes)  
checkInterval(only if checkType=1),



# tud.iir.news

## Class FeedClassifier

java.lang.Object

└--tud.iir.news.FeedClassifier

---

public class **FeedClassifier**  
extends java.lang.Object

The FeedClassifier classifies a feed in terms of their update intervals.

**Author:**

David Urbansky

---

## Fields

### CLASS\_UNKNOWN

public static final int **CLASS\_UNKNOWN**

feed class cannot be determined (feed not reachable)  
Constant value: 0

---

### CLASS\_ZOMBIE

public static final int **CLASS\_ZOMBIE**

feed was active but is not anymore  
Constant value: 1

---

### CLASS\_SPONTANUOUS

public static final int **CLASS\_SPONTANUOUS**

feed posts appear not often and at different intervals  
Constant value: 2

---

### CLASS\_SLICED

public static final int **CLASS\_SLICED**

feed posts are done at daytime with a longer gap at night  
Constant value: 3

---

### CLASS\_CONSTANT

public static final int **CLASS\_CONSTANT**

feed posts are 24/7 at a similar interval  
Constant value: 4

---

### CLASS\_CHUNKED

public static final int **CLASS\_CHUNKED**

---

(continued from last page)

all posts in the feed are updated together at a certain time  
Constant value: 5

---

## CLASS\_ON\_THE\_FLY

```
public static final int CLASS_ON_THE_FLY
```

all post entries are generated at request time  
Constant value: 6

## Constructors

### FeedClassifier

```
public FeedClassifier()
```

## Methods

### classify

```
public static int classify(java.lang.String feedURL)
```

---

### classify

```
public static int classify(java.lang.String feedURL,  
    FeedStore feedStore)
```

Classify a feed by its given URL.

**Parameters:**

`feedURL` - The URL of the feed.

**Returns:**

The class of the feed.

---

### classify

```
public static int classify(Feed feed)
```

---

### getClassName

```
public java.lang.String getClassName(int classID)
```

Get the name of the feed's class.

**Parameters:**

`classID` - The integer value of the class.

**Returns:**

The name of the class.

---

(continued from last page)

## main

```
public static void main(java.lang.String[] args)
```

### Parameters:

args

## tud.iir.news

# Class FeedContentClassifier

```
java.lang.Object
└--tud.iir.news.FeedContentClassifier
```

---

```
public class FeedContentClassifier
extends java.lang.Object
```

---

## Constructors

### FeedContentClassifier

```
public FeedContentClassifier()
```

---

### FeedContentClassifier

```
public FeedContentClassifier(FeedStore store)
```

---

## Methods

### determineFeedTextType

```
public int determineFeedTextType(java.lang.String feedUrl)
```

---

### determineFeedTextType

```
public int determineFeedTextType(Feed feed)
```

Try to determine the extent of text within a feed. We distinguish between no text [Feed.TEXT\\_TYPE\\_NONE](#), partial text [Feed.TEXT\\_TYPE\\_PARTIAL](#) and full text [Feed.TEXT\\_TYPE\\_FULL](#).

#### Parameters:

```
syndFeed
feedUrl
```

#### Returns:

---

### getReadableFeedTextType

```
public java.lang.String getReadableFeedTextType(int i)
```

---

(continued from last page)

## main

```
public static void main(java.lang.String[] args)
```

# tud.iir.news

## Class FeedDatabase

```
java.lang.Object
└-- tud.iir.news.FeedDatabase
```

All Implemented Interfaces:

[FeedStore](#)

---

```
public class FeedDatabase
extends java.lang.Object
implements FeedStore
```

The FeedDatabase is an implementation of the FeedStore that stores feeds and entries in a relational database. TODO change schema to InnoDB?

Author:

Philipp Katz, David Urbansky, klemens.muthmann@googlemail.com

## Methods

### getInstance

```
public static FeedDatabase getInstance()
```

---

### addFeed

```
public boolean addFeed(Feed feed)
```

---

### updateFeed

```
public boolean updateFeed(Feed feed)
```

---

### getFeedPostDistribution

```
public java.util.Map getFeedPostDistribution(Feed feed)
```

---

### updateFeedPostDistribution

```
public void updateFeedPostDistribution(Feed feed,
    java.util.Map postDistribution)
```

---

(continued from last page)

---

## changeCheckApproach

```
public void changeCheckApproach()
```

When the check approach is switched we need to reset learned and calculated values such as check intervals, checks, lastHeadlines etc.

---

## getFeeds

```
public java.util.List getFeeds()
```

---

## getFeedByUrl

```
public Feed getFeedByUrl(java.lang.String feedUrl)
```

---

## getFeedById

```
public Feed getFeedById(int feedID)
```

---

## addFeedEntry

```
public boolean addFeedEntry(Feed feed,  
                             FeedEntry entry)
```

---

## getFeedEntryByRawId

```
public FeedEntry getFeedEntryByRawId(java.lang.String rawId)
```

---

## getFeedEntryByRawId

```
public FeedEntry getFeedEntryByRawId(int feedId,  
                                       java.lang.String rawId)
```

---

## getFeedEntryById

```
public FeedEntry getFeedEntryById(int id)
```

---

## getFeedEntries

```
public java.util.List getFeedEntries(int limit,  
                                       int offset)
```

Get the specified count of feed entries, starting at offset.

**Parameters:**

(continued from last page)

```
limit  
offset
```

**Returns:**

---

## getFeedEntries

```
public java.util.List getFeedEntries(java.lang.String sqlQuery)
```

Get FeedEntries by using a custom SQL query. The SELECT part must contain all appropriate columns with their names from the feed\_entries table.

**Parameters:**

```
sqlQuery
```

**Returns:**

---

## getFeedEntriesForEvaluation

```
public java.util.List getFeedEntriesForEvaluation(java.lang.String sqlQuery)
```

---

## getTags

```
public java.util.List getTags(FeedEntry entry)
```

Get tags for specified FeedEntry. Result as sorted descendingly.

**Parameters:**

```
entry
```

**Returns:**

---

## assignTags

```
public void assignTags(FeedEntry entry,  
    java.util.List tags)
```

---

## getFeedEntryIdsTaggedAs

```
public java.util.Set getFeedEntryIdsTaggedAs(java.lang.String tag)
```

---

## deleteFeedEntryById

```
public void deleteFeedEntryById(int id)
```

---



(continued from last page)

## clearFeedTables

```
public void clearFeedTables()
```

---

## main

```
public static void main(java.lang.String[] args)
```

# tud.iir.news

## Class FeedDiscovery

```
java.lang.Object
|
+--tud.iir.news.FeedDiscovery
```

---

```
public class FeedDiscovery
extends java.lang.Object
```

FeedDiscovery works like the following:

1. Query search engine with some terms (I use Yahoo, as I can get large amounts of results)
2. Get root URLs for hits
3. Check page for feeds using RSS/Atom autodiscovery feature

**Author:**

Philipp Katz, David Urbansky

---

## Constructors

### FeedDiscovery

```
public FeedDiscovery()
```

traffic counter TODO use crawler downloadSize instead?

## Methods

### setResultFilePath

```
public void setResultFilePath(java.lang.String resultFilePath)
```

---

### getResultFilePath

```
public java.lang.String getResultFilePath()
```

---

### setWriteResultFileContinuously

```
public void setWriteResultFileContinuously(boolean writeResultFileOnTheFly)
```

---

### isWriteResultFileContinuously

```
public boolean isWriteResultFileContinuously()
```

---

(continued from last page)

## addQuery

```
public void addQuery(java.lang.String query)
```

Add a query for the search engine.

**Parameters:**

`query` - The query to add.

---

## addQueries

```
public void addQueries(java.util.Collection queries)
```

Add queries for the search engine.

**Parameters:**

`queries` - A collection of queries.

---

## discoverFeeds

```
public java.util.List discoverFeeds(java.lang.String pageUrl)
```

Discovers feed links in supplied page URL.

**Parameters:**

`pageUrl`

**Returns:**

list of discovered feed URLs, empty list if no feeds are available, `null` if page could not be parsed.

---

## findFeeds

```
public void findFeeds()
```

Find feeds in all pages on the sites tack. We use threading here which yields in much faster results.

---

## getFeeds

```
public java.util.Collection getFeeds()
```

Returns URLs of discovered feeds.

**Returns:**

---

## saveToFile

```
public void saveToFile()
```

Saves the discovered feeds to a file.

**Parameters:**

`resultFile` - The file where the feeds should be saved to.

---

(continued from last page)

## setDebugDump

```
public void setDebugDump(boolean debugDump)
```

Dump all XML files.

**Parameters:**

debugDump

---

## setMaxThreads

```
public void setMaxThreads(int maxThreads)
```

Set max number of concurrent autodiscovery requests.

**Parameters:**

maxThreads

---

## setResultLimit

```
public void setResultLimit(int resultLimit)
```

Limit the number of results for each query.

**Parameters:**

resultLimit - The number of websites to query. This does not necessarily mean that we get totalResults of feeds per query, as some sites do not offer a feed and some offer multiple feeds.

---

## addIgnore

```
public boolean addIgnore(java.lang.String ignore)
```

Add to entry ignore list. Any feed url containing this string will be ignored.

**Parameters:**

ignore

**Returns:**

---

## setIgnores

```
public void setIgnores(java.util.Collection ignores)
```

---

## setOnlyPreferred

```
public void setOnlyPreferred(boolean onlyPreferred)
```

Disable this option, to extract *all* available feeds on each webpage. Elsewise we only extract the *preferred* feed, which means the first one mentioned on the page.

**Parameters:**

allFeeds

**See Also:**

<http://tools.ietf.org/id/draft-snell-atompub-autodiscovery-00.txt>

---

---

## isOnlyPreferred

```
public boolean isOnlyPreferred()
```

---

---

## setSearchEngine

```
public void setSearchEngine(int searchEngine)
```

---

---

## getSearchEngine

```
public int getSearchEngine()
```

---

---

## getStatistics

```
public java.lang.String getStatistics()
```

Returns some statistics about the dicoverly process.

Returns:

---

---

## setCombineQueries

```
public void setCombineQueries(boolean combineQueries)
```

---

---

## isCombineQueries

```
public boolean isCombineQueries()
```

---

---

## main

```
public static void main(java.lang.String[] args)
```

---

# tud.iir.news

## Class FeedDiscoveryCallback

```
java.lang.Object
└--tud.iir.news.FeedDiscoveryCallback
```

All Implemented Interfaces:

[CrawlerCallback](#)

---

```
public class FeedDiscoveryCallback
extends java.lang.Object
implements CrawlerCallback
```

This class is used as a callback to automatically detect news feeds on pages which are downloaded with the [Crawler](#). Discovered feed URLs are written into a text file. This is singleton as we have potentially multiple Crawler instances, but writing to the list must be coordinated. See feeds.conf for options concerning the discovery.

**Author:**

Philipp Katz

---

## Methods

### getInstance

```
public static FeedDiscoveryCallback getInstance()
```

**Returns:**

Singleton of [FeedDiscoveryCallback](#) which is shared among all [Crawler](#) instances.

---

### crawlerCallback

```
public void crawlerCallback(org.w3c.dom.Document document)
```

---

### main

```
public static void main(java.lang.String[] args)
```

# tud.iir.news

## Class FeedEntry

```
java.lang.Object
└--tud.iir.news.FeedEntry
```

---

```
public class FeedEntry
extends java.lang.Object
```

Represents a news entry within a feed ([Feed](#)).

**Author:**

Philipp Katz, David Urbansky

---

## Constructors

### FeedEntry

```
public FeedEntry()
```

## Methods

### getId

```
public int getId()
```

---

### setId

```
public void setId(int id)
```

---

### getFeedId

```
public int getFeedId()
```

---

### setFeedId

```
public void setFeedId(int feedId)
```

---

### getTitle

```
public java.lang.String getTitle()
```

---

---

## setTitle

```
public void setTitle(java.lang.String title)
```

---

---

## getLink

```
public java.lang.String getLink()
```

---

---

## setLink

```
public void setLink(java.lang.String link)
```

---

---

## getRawId

```
public java.lang.String getRawId()
```

---

---

## setRawId

```
public void setRawId(java.lang.String rawId)
```

---

---

## getPublished

```
public java.util.Date getPublished()
```

---

---

## setPublished

```
public void setPublished(java.util.Date published)
```

---

---

## getPublishedSQLTimestamp

```
public java.sql.Timestamp getPublishedSQLTimestamp()
```

---

---

## getAdded

```
public java.util.Date getAdded()
```

---

---

## setAdded

```
public void setAdded(java.util.Date added)
```

---



(continued from last page)

---

## getAddedSQLTimestamp

```
public java.sql.Timestamp getAddedSQLTimestamp()
```

---

## getEntryText

```
public java.lang.String getEntryText()
```

---

## setEntryText

```
public void setEntryText(java.lang.String entryText)
```

---

## getPageText

```
public java.lang.String getPageText()
```

---

## setPageText

```
public void setPageText(java.lang.String pageText)
```

---

## getText

```
public java.lang.String getText()
```

Get entry's text, either (preferably) from the page or from the feed. Never return `null`.

**Returns:**

---

## getFeatures

```
public java.util.SortedMap getFeatures()
```

---

## setFeatures

```
public void setFeatures(java.util.SortedMap features)
```

---

## getFeature

```
public double getFeature(java.lang.String key)
```

---

---

## putFeature

```
public void putFeature(java.lang.String key,  
                      double value)
```

---

## toString

```
public java.lang.String toString()
```

## tud.iir.news

# Class FeedPostStatistics

```
java.lang.Object
└-- tud.iir.news.FeedPostStatistics
```

---

```
public class FeedPostStatistics
    extends java.lang.Object
```

Capture some statistics about the posts of a feed.

**Author:**  
David Urbansky

---

## Constructors

### FeedPostStatistics

```
public FeedPostStatistics(java.util.List feedEntries)
```

## Methods

### getTimeRange

```
public long getTimeRange()
```

---

### getTimeDifferenceToNewestPost

```
public long getTimeDifferenceToNewestPost()
```

---

### getPostDistribution

```
public java.util.Map getPostDistribution()
```

---

### setPostDistribution

```
public void setPostDistribution(java.util.Map postDistribution)
```

---

### getTimeOldestPost

```
public long getTimeOldestPost()
```

---

---

## setTimeOldestPost

```
public final void setTimeOldestPost(long timeOldestPost)
```

---

---

## getTimeNewestPost

```
public long getTimeNewestPost()
```

---

---

## setTimeNewestPost

```
public final void setTimeNewestPost(long timeNewestPost)
```

---

---

## getMedianPostGap

```
public long getMedianPostGap()
```

---

---

## setMedianPostGap

```
public final void setMedianPostGap(long medianPostGap)
```

---

---

## setPostGapStandardDeviation

```
public final void setPostGapStandardDeviation(long postGapStandardDeviation)
```

---

---

## getPostGapStandardDeviation

```
public long getPostGapStandardDeviation()
```

---

---

## setLongestPostGap

```
public void setLongestPostGap(long longestPostGap)
```

---

---

## getLongestPostGap

```
public long getLongestPostGap()
```

---

---

## toString

```
public java.lang.String toString()
```

---

(continued from last page)

## tud.iir.news

# Class FeedProcessingAction

java.lang.Object

└-- tud.iir.news.FeedProcessingAction

---

public abstract class **FeedProcessingAction**  
extends java.lang.Object

---

## Fields

### arguments

public java.lang.Object **arguments**

## Constructors

### FeedProcessingAction

public **FeedProcessingAction**()

---

### FeedProcessingAction

public **FeedProcessingAction**(java.lang.Object[] parameters)

---

## Methods

### performAction

public abstract void **performAction**([Feed](#) feed)

# tud.iir.news

## Interface FeedStore

All Known Implementing Classes:

[FeedDatabase](#), [FeedStoreDummy](#)

---

public interface **FeedStore**  
extends

The FeedStore is an interface for feed stores such as databases or file indices.

**Author:**

Philipp Katz, David Urbansky

---

## Methods

### addFeed

```
public boolean addFeed(Feed feed)
```

Add a new feed if its feedURL does not yet exist.

**Parameters:**

`feed` - The feed to add.

**Returns:**

true if feed was added successfully

---

### updateFeed

```
public boolean updateFeed(Feed feed)
```

Update a feed if its feedURL already exists.

**Parameters:**

`feed` - The feed to update.

**Returns:**

true if feed was updated successfully

---

### getFeeds

```
public java.util.List getFeeds()
```

Get all feeds.

**Returns:**

A list of all feeds from the store.

---

### getFeedByUrl

```
public Feed getFeedByUrl(java.lang.String feedUrl)
```

Get a feed by its feedUrl.

---

(continued from last page)

**Parameters:**

feedUrl

**Returns:**

the Feed with specified feedUrl, null if Feed does not exist.

---

## addFeedEntry

```
public boolean addFeedEntry(Feed feed,  
    FeedEntry entry)
```

If it does not yet exist, add an entry to an existing feed.

**Parameters:**feed  
entry

---

## getFeedEntryByRawId

```
public FeedEntry getFeedEntryByRawId(java.lang.String rawId)
```

**Deprecated.** use [getFeedEntryByRawId\(int, String\)](#) instead.

Get an entry by its rawId.

**Parameters:**

rawId

**Returns:**

the FeedEntry with specified rawId, null if FeedEntry does not exist.

---

## getFeedEntryByRawId

```
public FeedEntry getFeedEntryByRawId(int feedId,  
    java.lang.String rawId)
```

Get an entry for a specific feed by its rawId.

**Parameters:**feedId  
rawId**Returns:**

the FeedEntry with specified rawId, null if FeedEntry does not exist.

---

## getFeedById

```
public Feed getFeedById(int feedID)
```

---

## getFeedEntries

```
public java.util.List getFeedEntries(java.lang.String sqlQuery)
```

Get FeedEntries by using a custom SQL query. The SELECT part must contain all appropriate columns with their names from the feed\_entries table.

**Parameters:**



sqlQuery

(continued from last page)

**Returns:**

---

## getFeedEntryIdsTaggedAs

```
public java.util.Set getFeedEntryIdsTaggedAs(java.lang.String tag)
```

# tud.iir.news

## Class FeedStoreDummy

```
java.lang.Object
└--tud.iir.news.FeedStoreDummy
```

All Implemented Interfaces:  
[FeedStore](#)

---

```
public class FeedStoreDummy
extends java.lang.Object
implements FeedStore
```

Dummy/mock class which can be used instead of "real" database for testing purposes.

**Author:**  
Philipp Katz

---

## Constructors

### FeedStoreDummy

```
public FeedStoreDummy()
```

## Methods

### addFeedEntry

```
public boolean addFeedEntry(Feed feed,
FeedEntry entry)
```

---

### addFeed

```
public boolean addFeed(Feed feed)
```

---

### getFeedByUrl

```
public Feed getFeedByUrl(java.lang.String feedUrl)
```

---

### getFeeds

```
public java.util.List getFeeds()
```

---

(continued from last page)

## getFeedEntryByRawId

```
public FeedEntry getFeedEntryByRawId(java.lang.String rawId)
```

---

## updateFeed

```
public boolean updateFeed(Feed feed)
```

---

## getFeedById

```
public Feed getFeedById(int feedID)
```

---

## getFeedEntryByRawId

```
public FeedEntry getFeedEntryByRawId(int feedId,  
                                       java.lang.String rawId)
```

---

## getFeedEntryIdsTaggedAs

```
public java.util.Set getFeedEntryIdsTaggedAs(java.lang.String tag)
```

---

## getFeedEntries

```
public java.util.List getFeedEntries(java.lang.String sqlQuery)
```

---

# tud.iir.news Class Helper

```
java.lang.Object
|
|--tud.iir.news.Helper
```

---

```
public class Helper
extends java.lang.Object
```

Various more or less feed specific helper functions. TODO most of these methods can be moved to the global Helper classes. TODO move methods, which are used by PageContentExtractor to global HTMLHelper!

**Author:**  
Philipp Katz

---

## Methods

### xmlToString

```
public static java.lang.String xmlToString(org.w3c.dom.Node node)
```

---

### xmlToString

```
public static java.lang.String xmlToString(org.w3c.dom.Node node,
        boolean removeWhitespace,
        boolean prettyPrint)
```

Converts a DOM Node or Document into a String. TODO removing whitespace does not work with documents from the Crawler/Neko?

**Parameters:**

node  
removeWhitespace - whether to remove superfluous whitespace outside of tags.  
prettyPrint - wheter to nicely indent the result.

**Returns:**

String representation of the supplied Node, empty String in case of errors.

---

### removeWhitespace

```
public static org.w3c.dom.Node removeWhitespace(org.w3c.dom.Node node)
```

Remove unnecessary whitespace from DOM nodes.  
<http://stackoverflow.com/questions/978810/how-to-strip-whitespace-only-text-nodes-from-a-dom-before-serialization>

**Parameters:**

node

**Returns:**

---

(continued from last page)

## writeXmlDump

```
public static void writeXmlDump(org.w3c.dom.Node node,  
    java.lang.String filename)
```

---

## getFirstWords

```
public static java.lang.String getFirstWords(java.lang.String string,  
    int num)
```

Shorten a String; returns the first num words.

**Parameters:**

string  
num

**Returns:**

---

## countOccurences

```
public static int countOccurences(java.lang.String text,  
    java.lang.String pattern,  
    boolean ignoreCase)
```

Count number of occurences of pattern withing text. TODO this will fail if pattern contains RegEx metacharacters. Need to escape.

**Parameters:**

text  
pattern  
ignoreCase

**Returns:**

---

## stringToXml

```
public static org.w3c.dom.Document stringToXml(java.lang.String input)
```

Converts a String representation with XML markup to DOM Document. Returns an empty Document if parsing failed.

**Parameters:**

input

**Returns:**

---

## getOuterXml

```
public static java.lang.String getOuterXml(org.w3c.dom.Node node)
```

Returns a String representation of the supplied Node, including the Node itself, like outerHTML in JavaScript/DOM. <http://chicknet.blogspot.com/2007/05/outerxml-for-java.html>

**Parameters:**

node

(continued from last page)

**Returns:**

---

## getInnerXml

```
public static java.lang.String getInnerXml(org.w3c.dom.Node node)
```

Returns a String representation of the supplied Node, excluding the Node itself, like innerHTML in JavaScript/DOM.

**Parameters:**

node

**Returns:**

---

## createDocument

```
public static org.w3c.dom.Document createDocument()
```

Creates a new, empty DOM Document.

**Returns:**

---

## removeAll

```
public static void removeAll(org.w3c.dom.Node node,  
    short nodeType)
```

Removes all nodes with specified type from Node.

**Parameters:**

node

nodeType - for example Node.COMMENT\_NODE

---

## removeAll

```
public static void removeAll(org.w3c.dom.Node node,  
    short nodeType,  
    java.lang.String name)
```

Removes all nodes with specified type and name from Node.

**Parameters:**

node

nodeType - for example Node.COMMENT\_NODE

name

---

## cloneDocument

```
public static org.w3c.dom.Document cloneDocument(org.w3c.dom.Document document)
```

Creates a copy of a DOM Document. <http://stackoverflow.com/questions/279154/how-can-i-clone-an-entire-document-using-the-java-dom>

**Parameters:**

document

(continued from last page)

**Returns:**the cloned Document or `null` if cloning failed.

---

## getLevenshteinSim

```
public static float getLevenshteinSim(java.lang.String s1,  
                                       java.lang.String s2)
```

Calculates Levenshtein similarity between the strings.

**Parameters:**

s1  
s2

**Returns:**

similarity between 0 and 1 (inclusive).

---

## getLengthSim

```
public static float getLengthSim(java.lang.String s1,  
                                   java.lang.String s2)
```

Determine similarity based on String lengths. We can use this as threshold before even calculating Levenshtein similarity which is computationally expensive.

**Parameters:**

s1  
s2

**Returns:**

similarity between 0 and 1 (inclusive).

---

## getReadableBytes

```
public static java.lang.String getReadableBytes(long bytes)
```

Format number of bytes to human readable String using IEC binary unit prefixes, for example `getReadableBytes(48956748) -> 46.69 MiB`**Parameters:**

bytes

**Returns:**

---

## main

```
public static void main(java.lang.String[] args)
```

# tud.iir.news

## Class NewsAggregator

```
java.lang.Object
└-- tud.iir.news.NewsAggregator
```

---

```
public class NewsAggregator
extends java.lang.Object
```

NewsAggregator uses ROME library to fetch and parse feeds from the web. Feeds are stored persistently, aggregation method fetches new entries. TODO add a "lastSuccessfulAggregation" attribute to feed, so we can filter out obsolete feeds. TODO we should check if an entry was modified and update. TODO determine feed format for statistics? --> [https://rome.dev.java.net/apidocs/1\\_0/com/sun/syndication/feed/WireFeed.html#getFeedType\(\)](https://rome.dev.java.net/apidocs/1_0/com/sun/syndication/feed/WireFeed.html#getFeedType()) TODO add a general filter to ignore specific types of feeds, for example by language, count of entries, URL pattern, etc. <https://rome.dev.java.net/> \*

**Author:**  
Philipp Katz

---

## Constructors

### NewsAggregator

```
public NewsAggregator()
```

---

### NewsAggregator

```
public NewsAggregator(FeedStore store)
```

Used primarily for testing to set DummyFeedStore.

## Methods

### addFeed

```
public boolean addFeed(java.lang.String feedUrl)
```

Adds a new feed for aggregation.

**Parameters:**  
feedUrl

**Returns:**  
true, if feed was added.

---

### updateFeed

```
public boolean updateFeed(Feed feed)
```

---



(continued from last page)

## addFeeds

```
public int addFeeds(java.util.Collection feedUrls)
```

Add a Collection of feedUrls for aggregation. This process runs threaded. Use [setMaxThreads\(int\)](#) to set the maximum number of concurrently running threads.

**Parameters:**

feedUrls

**Returns:**

number of added feeds.

---

## addFeedsFromFile

```
public int addFeedsFromFile(java.lang.String filePath)
```

Add feeds from a supplied file. The file must contain a newline separated list of feed URLs.

**Parameters:**

fileName

**Returns:**

---

## aggregate

```
public int aggregate()
```

Do the aggregation process. New entries from all known feeds will be aggregated. Use [setMaxThreads\(int\)](#) to set the number of maximum parallel threads. TODO use Thread Pools? <http://developer.amd.com/documentation/articles/pages/1121200683.aspx>  
<http://www.ibm.com/developerworks/library/j-jtp0730.html>

**Returns:**

number of aggregated new entries.

---

## aggregateContinuously

```
public void aggregateContinuously(int waitMinutes)
```

Runs a continuous aggregation process. This is mainly intended for use as background process from the command line.

**Parameters:**

waitMinutes - the interval in seconds when the aggregation is done.

**Returns:**

---

## setMaxThreads

```
public void setMaxThreads(int maxThreads)
```

Sets the maximum number of parallel threads when aggregating or adding multiple new feeds.

**Parameters:**

maxThreads

---

---

## setDownloadPages

```
public void setDownloadPages(boolean downloadPages)
```

If enabled, we use [PageContentExtractor](#) to analyse feed type and to extract more text from feed entries with only partial text representations. Keep in mind that this causes heavy traffic and therfor takes a lot more time than a simple aggregation process from XML feeds only.

### Parameters:

downloadPages

---

## isDownloadPages

```
public boolean isDownloadPages()
```

---

## downloadFeed

```
public Feed downloadFeed(java.lang.String feedUrl)  
    throws NewsAggregatorException
```

Returns a feed and its entries from a specified feed URL. Use [Feed.getEntries\(\)](#) to get feed's entries.

### Parameters:

feedUrl

### Returns:

### Throws:

[NewsAggregatorException](#)

---

## main

```
public static void main(java.lang.String[] args)
```

Main method with command line interface.

### Parameters:

args

---

## tud.iir.news

# Class NewsAggregatorException

```
java.lang.Object
├-- java.lang.Throwable
│   └-- java.lang.Exception
│       └-- tud.iir.news.NewsAggregatorException
```

### All Implemented Interfaces:

java.io.Serializable

---

```
public class NewsAggregatorException
extends java.lang.Exception
```

---

## Constructors

### NewsAggregatorException

```
public NewsAggregatorException(java.lang.Throwable t)
```

### NewsAggregatorException

```
public NewsAggregatorException(java.lang.String string)
```

---

Package  
**tud.iir.normalization**

# tud.iir.normalization

## Class DateNormalizer

```
java.lang.Object
└--tud.iir.normalization.DateNormalizer
```

```
public class DateNormalizer
extends java.lang.Object
```

The DateNormalizer normalizes dates.

## Constructors

### DateNormalizer

```
public DateNormalizer()
```

## Methods

### normalizeDateFormat

```
public static java.lang.String normalizeDateFormat(java.util.Date date,
    java.lang.String format)
```

### normalizeDateFormat

```
public static java.lang.String normalizeDateFormat(java.lang.String dateString,
    java.lang.String format)
```

### normalizeDate

```
public static java.lang.String normalizeDate(java.lang.String dateString)
```

Normalize a given date to the format YYYY-MM-DD (UTC standard).

#### Parameters:

`dateString` - The date string.

#### Returns:

The normalized date in UTC standard.

### normalizeDate

```
public static java.lang.String normalizeDate(java.lang.String dateString,
    boolean fillTime)
```

## main

```
public static void main(java.lang.String[] args)
```

# tud.iir.normalization

## Class StringNormalizer

```
java.lang.Object
└--tud.iir.normalization.StringNormalizer
```

---

```
public class StringNormalizer
extends java.lang.Object
```

The string normalizer normalizes strings.

**Author:**

David Urbansky

---

## Constructors

### StringNormalizer

```
public StringNormalizer()
```

## Methods

### normalizeNumber

```
public static java.lang.String normalizeNumber(java.lang.String numberString)
```

Different number formats do not match if compared, thus they have to be normalized before. e.g. 40,000 = 40000 and 4.00 = 4.0 = 4 but 6,6 should be equal to 6.6

**Parameters:**

`numberString` - The string with a number.

**Returns:**

The normalized number as a string.

---

### main

```
public static void main(java.lang.String[] args)
```

**Parameters:**

`args`

---

# tud.iir.normalization

## Class UnitNormalizer

```
java.lang.Object
└--tud.iir.normalization.UnitNormalizer
```

---

```
public class UnitNormalizer
extends java.lang.Object
```

The UnitNormalizer normalizes units.

**Author:**  
David Urbansky

---

## Constructors

### UnitNormalizer

```
public UnitNormalizer()
```

## Methods

### isBigger

```
public static boolean isBigger(java.lang.String unitB,
                               java.lang.String unitS)
```

Returns true if unitB is bigger than units. e.g. hours > minutes and GB > MB

**Parameters:**

unitB - The bigger unit.  
unitS - The smaller unit.

**Returns:**

True if unitB is bigger than unitS.

---

### unitsSameType

```
public static boolean unitssameType(java.lang.String unit1,
                                     java.lang.String unit2)
```

Returns true if units are the same unit type (time,distance etc.). e.g. MB and GB are digital size, hours and minutes are time units

**Parameters:**

unit1 - The first unit.  
unit2 - The second unit.

**Returns:**

True if both units are the same type.

---



(continued from last page)

## unitLookup

```
public static double unitLookup(java.lang.String unit)
```

---

## handleSpecialFormat

```
public static double handleSpecialFormat(double number,  
    java.lang.String unitText,  
    int decimals)
```

Find special formats for combined values (well formed as "1 min 4 sec" are handled by `getNormalizedNumber`). 1m20s => 80s 1h2m20s => 3740s (1m:20s => 80s) 00:01:20 => 80s 1:20 => 80s 5'9" => 175.26cm 5'9" => 175.26cm

### Parameters:

number - The number.  
unitText - The text after the unit.

### Returns:

The combined value or -1 if number is not part of special format.

---

## getUnitTypeName

```
public static java.lang.String getUnitTypeName(java.lang.String string)
```

---

## getUnitType

```
public static int getUnitType(java.lang.String string)
```

---

## getNormalizedNumber

```
public static double getNormalizedNumber(java.lang.String unitText)  
    throws java.lang.NumberFormatException,  
    java.lang.NullPointerException
```

---

## getNormalizedNumber

```
public static double getNormalizedNumber(double number,  
    java.lang.String unitText)
```

---

## getNormalizedNumber

```
public static double getNormalizedNumber(double number,  
    java.lang.String unitText,  
    int decimals,  
    java.lang.String combinedSearchPreviousUnit)
```

---

(continued from last page)

## main

```
public static void main(java.lang.String[] args)
```

### Parameters:

args

---

Package  
**tud.iir.persistence**

# tud.iir.persistence

## Class DatabaseManager

java.lang.Object

└─tud.iir.persistence.DatabaseManager

---

public class **DatabaseManager**  
extends java.lang.Object

The DatabaseManager writes and reads data to the database.

**Author:**

David Urbansky, Christopher Friedrich, Philipp Katz, Martin Werner

---

## Methods

### getInstance

public static [DatabaseManager](#) **getInstance()**

Gets the single instance of DatabaseManager.

**Returns:**

single instance of DatabaseManager

---

### getConnection

public java.sql.Connection **getConnection()**

Return the connection.

**Returns:**

### updateOntology

public void **updateOntology()**

Update ontology.

---

### updateOntology

public void **updateOntology**(java.lang.String filePath)

Write the concepts and their attributes (defined in the ontology) in the database.

**Parameters:**

filePath - the file path

---

### updateOntology

public void **updateOntology**([KnowledgeManager](#) knowledgeManager)

Update ontology.

---

(continued from last page)

**Parameters:**

knowledgeManager - the knowledge manager

---

## loadOntology

```
public KnowledgeManager loadOntology()
```

Load ontology.

**Returns:**

the knowledge manager

---

## loadOntology

```
public KnowledgeManager loadOntology(java.lang.String filePath)
```

Load the ontology saved in the database into the KnowledgeManager. Update the ontology for the database first from the owl ontology.

**Parameters:**

filePath - the file path

**Returns:**

the knowledge manager

---

## loadEntities

```
public java.util.ArrayList loadEntities(Concept concept,  
    int number,  
    int offset,  
    boolean continueFromLastExtraction)
```

Load entities (names and lastSearched only) for a specific concept.

**Parameters:**

concept - the concept

number - Number of Entities to return.

offset - An offset value.

continueFromLastExtraction - the continue from last extraction

**Returns:**

An array of entities.

---

## loadConcepts

```
public java.util.ArrayList loadConcepts()
```

Load concepts.

**Returns:**

the array list

---

## loadEvaluationEntities

```
public java.util.ArrayList loadEvaluationEntities(Concept concept)
```

Load evaluation entities.

**Parameters:**

(continued from last page)

concept - the concept

**Returns:**  
the array list

---

## loadEntity

```
public Entity loadEntity(int entityID)
```

Load entity.

**Parameters:**  
entityID - the entity id

**Returns:**  
the entity

---

## saveExtractions

```
public void saveExtractions(KnowledgeManager knowledgeManager)
```

Save instance knowledge (entities, their facts(also MIOs), their snippets and their sources). If entries exist, link them.

**Parameters:**  
knowledgeManager - The knowledgeManager.

---

## getSeeds

```
public java.util.ArrayList getSeeds(Concept concept,  
int maxNumber)
```

Gets the seeds.

**Parameters:**  
concept - the concept  
maxNumber - the max number

**Returns:**  
the seeds

---

## getPMI

```
public double getPMI(int entityID,  
int attributeID)
```

Gets the pMI.

**Parameters:**  
entityID - the entity id  
attributeID - the attribute id

**Returns:**  
the pMI

---

## getBenchmarkPMIs

```
public java.util.HashMap getBenchmarkPMIs()
```

(continued from last page)

Gets the benchmark pm is.

**Returns:**

the benchmark pm is

## updateExtractionStatus

```
public void updateExtractionStatus(int extractionPhase,
    int progress,
    java.lang.StringBuilder logExcerpt,
    long downloadedBytes)
```

\*\*\*\*\*  
reading and writing from the database  
\*\*\*\*\*.

**Parameters:**

extractionPhase - the extraction phase  
progress - the progress  
logExcerpt - the log excerpt  
downloadedBytes - the downloaded bytes

## getExtractionStatusDownloadedBytes

```
public long getExtractionStatusDownloadedBytes()
```

Gets the extraction status downloaded bytes.

**Returns:**

the extraction status downloaded bytes

## setTestField

```
public void setTestField(int entityID,
    boolean test)
```

Set the test field in training\_samples database for a certain entity.

**Parameters:**

entityID - the entity id  
test - the test

## getEntitiesForSource

```
public java.util.HashSet getEntitiesForSource(int sourceID)
```

Gets the entities for source.

**Parameters:**

sourceID - the source id

**Returns:**

the entities for source

## getEntitiesForExtractionType

```
public java.util.HashSet getEntitiesForExtractionType(int extractionType,
    Concept concept)
```

Gets the entities for extraction type.

(continued from last page)

**Parameters:**

extractionType - the extraction type  
concept - the concept

**Returns:**

the entities for extraction type

---

## getExtractionTypesForSource

```
public java.util.HashSet getExtractionTypesForSource(int sourceID,  
    Concept concept)
```

Gets the extraction types for source.

**Parameters:**

sourceID - the source id  
concept - the concept

**Returns:**

the extraction types for source

---

## getSourcesForExtractionType

```
public java.util.HashSet getSourcesForExtractionType(int extractionType,  
    Concept concept)
```

Gets the sources for extraction type.

**Parameters:**

extractionType - the extraction type  
concept - the concept

**Returns:**

the sources for extraction type

---

## addQAs

```
public void addQAs(java.util.List qas)
```

Adds the q as.

**Parameters:**

qas - the qas

---

## addAttributeSynonym

```
public int addAttributeSynonym(int attributeID1,  
    int attributeID2,  
    double trust)
```

Add an attribute synonym pair.

**Parameters:**

attributeID1 - Attribute id 1.  
attributeID2 - Attribute id 2.  
trust - The trust in the connection.

**Returns:**



(continued from last page)

The id of the added attribute synonym.

---

## calculateAttributeSynonymTrust

```
public double calculateAttributeSynonymTrust(Attribute attribute1,  
                                             Attribute attribute2)
```

Calculate attribute synonym trust.

### Parameters:

attribute1 - the attribute1  
attribute2 - the attribute2

### Returns:

the double

---

## getEntityName

```
public java.lang.String getEntityName(int entityID)
```

Gets the entity name.

### Parameters:

entityID - the entity id

### Returns:

the entity name

---

## getEntityIDsByName

```
public java.util.HashSet getEntityIDsByName(java.lang.String entityName)
```

Gets the entity i ds by name.

### Parameters:

entityName - the entity name

### Returns:

the entity i ds by name

---

## addAssessmentInstance

```
public void addAssessmentInstance(int conceptID,  
                                   int entityID,  
                                   int classValue)
```

Adds the assessment instance.

### Parameters:

conceptID - the concept id  
entityID - the entity id  
classValue - the class value

---

## addFact

```
public int addFact(FactValue factValue,  
                   int entityID,  
                   int attributeID)
```

(continued from last page)

Add a fact value (the fact in the facts table and the value in the values table).

**Parameters:**

factValue - The fact value.  
entityID - The entity id.  
attributeID - The attribute id.

**Returns:**

The id of the added fact.

---

## addFact

```
public int addFact(FactValue factValue,  
                  int entityID,  
                  int attributeID,  
                  double trust)
```

Add a factValue (especially for MIOs).

**Parameters:**

factValue - the fact value  
entityID - the entity id  
attributeID - the attribute id  
trust - the trust

**Returns:**

the int

---

## getSourceURL

```
public java.lang.String getSourceURL(int sourceID)
```

Gets the source url.

**Parameters:**

sourceID - the source id

**Returns:**

the source url

---

## getSnippetID

```
public int getSnippetID(Snippet snippet)
```

Gets the snippet id.

**Parameters:**

snippet - the snippet

**Returns:**

the snippet id

---

## snippetExists

```
public boolean snippetExists(Snippet snippet)
```

Snippet exists.

**Parameters:**

(continued from last page)

snippet - the snippet

**Returns:**  
true, if successful

---

## getLastInsertID

```
public int getLastInsertID()
```

Gets the last insert id.

**Returns:**  
the last insert id

---

## getConceptID

```
public int getConceptID(java.lang.String conceptName)
```

Gets the concept id.

**Parameters:**  
conceptName - the concept name

**Returns:**  
the concept id

---

## getTotalConceptsNumber

```
public int getTotalConceptsNumber()
```

Gets the total concepts number.

**Returns:**  
the total concepts number

---

## getTotalAttributesNumber

```
public int getTotalAttributesNumber()
```

Gets the total attributes number.

**Returns:**  
the total attributes number

---

## getTotalEntitiesNumber

```
public int getTotalEntitiesNumber()
```

Gets the total entities number.

**Returns:**  
the total entities number

---

## getTotalEntitiesNumber

```
public int getTotalEntitiesNumber(java.lang.String conceptName)
```

Gets the total entities number.

---

(continued from last page)

**Parameters:**`conceptName` - the concept name**Returns:**the total entities number

---

## getTotalFactsNumber

```
public int getTotalFactsNumber()
```

Total number of facts (only one per entity-attribute).

**Returns:**The total number of facts.

---

## getTotalFactsNumber

```
public int getTotalFactsNumber(java.lang.String conceptName)
```

Gets the total facts number.

**Parameters:**`conceptName` - the concept name**Returns:**the total facts number

---

## getTotalSourcesNumber

```
public int getTotalSourcesNumber()
```

Gets the total sources number.

**Returns:**the total sources number

---

## runQuery

```
public java.sql.ResultSet runQuery(java.lang.String query)
```

Run query.

**Parameters:**`query` - the query**Returns:**the result set

---

## runQuery

```
public java.sql.ResultSet runQuery(java.sql.PreparedStatement statement)
```

Run query.

**Parameters:**`statement` - the statement

---

(continued from last page)

**Returns:**  
the result set

---

## runQuery

```
public java.sql.ResultSet runQuery(java.lang.String query,  
    java.lang.String text)
```

Run query.

**Parameters:**  
query - the query  
text - the text

**Returns:**  
the result set

---

## runQuery

```
public java.sql.ResultSet runQuery(java.lang.String query,  
    java.lang.String[] texts)
```

Run query.

**Parameters:**  
query - the query  
texts - the texts

**Returns:**  
the result set

---

## runUpdate

```
public int runUpdate(java.sql.PreparedStatement preparedStatement)
```

Execute a prepared statement.

**Parameters:**  
preparedStatement - The prepared statement.

**Returns:**  
the int

---

## runUpdate

```
public int runUpdate(java.lang.String update)
```

Run update.

**Parameters:**  
update - the update

**Returns:**  
the int

---

(continued from last page)

## runUpdate

```
public int runUpdate(java.lang.String update,  
                    java.lang.String text)
```

Run update.

**Parameters:**

update - the update

text - the text

**Returns:**

the int

---

## runUpdate

```
public int runUpdate(java.lang.String update,  
                    java.lang.String[] texts)
```

Run update.

**Parameters:**

update - the update

texts - the texts

**Returns:**

the int

---

## cleanUnusedOntologyElements

```
public void cleanUnusedOntologyElements()
```

Deletes all domains, concepts, attributes that are not in the ontology anymore (foreign key cascade). It also deletes all facts etc. that refer to them (trigger / lookup).

---

## clearCompleteDatabase

```
public void clearCompleteDatabase()
```

Clear complete database.

---

## testProcedure

```
public void testProcedure()
```

Test procedure.

---

## getWorstIndices

```
public void getWorstIndices()
```

Sql script to grab the worst performing indexes in the whole server. Source:  
<http://forge.mysql.com/tools/tool.php?id=85>

---

## main

```
public static void main(java.lang.String[] args)  
    throws java.lang.Exception
```

---

(continued from last page)

The main method.

**Parameters:**

`args` - the arguments

**Throws:**

`Exception` - the exception

# tud.iir.persistence

## Class DictionaryDBIndexH2

```
java.lang.Object
├── tud.iir.persistence.DictionaryIndex
│   └── tud.iir.persistence.DictionaryDBIndexH2
```

```
public class DictionaryDBIndexH2
extends DictionaryIndex
```

## Constructors

### DictionaryDBIndexH2

```
public DictionaryDBIndexH2(java.lang.String dbName,
                           java.lang.String dbUsername,
                           java.lang.String dbPassword,
                           java.lang.String indexPath)
```

### DictionaryDBIndexH2

```
public DictionaryDBIndexH2(java.lang.String dbName,
                           java.lang.String dbUsername,
                           java.lang.String dbPassword)
```

### DictionaryDBIndexH2

```
public DictionaryDBIndexH2(java.lang.String dbName,
                           java.lang.String dbUsername,
                           java.lang.String dbPassword,
                           boolean inMemoryMode)
```

Constructor with the choice of using a in-memory data base or writing it to disk and connects to the data base

#### Parameters:

`dbName` - The name of the data base. If it does not exist, it will be created

`dbUsername` - The user name for the data base.

`dbPassword` - The user's password.

`inMemoryMode` - If true, the db will be kept in memory until the virtual machine is terminated, if false, db is serialized to disk.

## Methods

### empty

```
public void empty()
```



## read

```
public CategoryEntries read(java.lang.String word)
```

---

## read1

```
public CategoryEntries read1(java.lang.String word)
```

Read the word from the unnormalized table with all information (faster).

**Parameters:**

`word` - The word to look up.

**Returns:**

The category entries for the word.

---

## read3

```
public CategoryEntries read3(java.lang.String word)
```

Read the word from the 3 normalized tables (more space efficient).

**Parameters:**

`word` - The word to look up.

**Returns:**

The category entries for the word.

---

## update

```
public void update(java.lang.String word,  
    CategoryEntries categoryEntries)
```

---

## update

```
public void update(java.lang.String word,  
    CategoryEntry categoryEntry)
```

---

## write

```
public void write(java.lang.String word,  
    CategoryEntries categoryEntries)
```

---

## write

```
public void write(java.lang.String word,  
    CategoryEntry categoryEntry)
```

---

---

## write3

```
public void write3(java.lang.String word,  
    CategoryEntries categoryEntries)
```

Write a word with its category entries into the dictionary (3 tables, more space efficient).

### Parameters:

word - The word to write.

categoryEntries - The category entries for the word.

---

## getDbType

```
public java.lang.String getDbType()
```

---

## setDbType

```
public void setDbType(java.lang.String dbType)
```

---

## getDbDriver

```
public java.lang.String getDbDriver()
```

---

## setDbDriver

```
public void setDbDriver(java.lang.String dbDriver)
```

---

## getDbHost

```
public java.lang.String getDbHost()
```

---

## setDbHost

```
public void setDbHost(java.lang.String dbHost)
```

---

## getDbPort

```
public java.lang.String getDbPort()
```

---

## setDbPort

```
public void setDbPort(java.lang.String dbPort)
```

---

(continued from last page)

---

## getDbName

```
public java.lang.String getDbName()
```

---

## setDbName

```
public void setDbName(java.lang.String dbName)
```

---

## getDbUsername

```
public java.lang.String getDbUsername()
```

---

## setDbUsername

```
public void setDbUsername(java.lang.String dbUsername)
```

---

## getDbPassword

```
public java.lang.String getDbPassword()
```

---

## setDbPassword

```
public void setDbPassword(java.lang.String dbPassword)
```

---

## isFastMode

```
public boolean isFastMode()
```

---

## setFastMode

```
public void setFastMode(boolean fastMode)
```

---

## close

```
public void close()
```

---

(continued from last page)

## openReader

```
public boolean openReader()
```

---

## openWriter

```
public void openWriter()
```

---

## isInMemoryMode

```
public boolean isInMemoryMode()
```

The mode this data base is working in.

**Returns:**

If true, the db is kept in memory until the virtual machine is closed, if false, db is serialized to disk.

---

## setInMemoryMode

```
public void setInMemoryMode(boolean inMemoryMode)
```

# tud.iir.persistence

## Class DictionaryDBIndexMySQL

java.lang.Object

```

  |
+- tud.iir.persistence.DictionaryIndex
  |
+- tud.iir.persistence.DictionaryDBIndexMySQL

```

```

public class DictionaryDBIndexMySQL
extends DictionaryIndex

```

## Constructors

### DictionaryDBIndexMySQL

```

public DictionaryDBIndexMySQL( java.lang.String dbName,
                               java.lang.String dbUsername,
                               java.lang.String dbPassword,
                               java.lang.String indexPath)

```

### DictionaryDBIndexMySQL

```

public DictionaryDBIndexMySQL( java.lang.String dbName,
                               java.lang.String dbUsername,
                               java.lang.String dbPassword)

```

## Methods

### empty

```

public void empty()

```

### read

```

public CategoryEntries read( java.lang.String word)

```

### read1

```

public CategoryEntries read1( java.lang.String word)

```

Read the word from the unnormalized table with all information (faster).

#### Parameters:

word - The word to look up.

(continued from last page)

**Returns:**

The category entries for the word.

---

**read3**

```
public CategoryEntries read3(java.lang.String word)
```

Read the word from the 3 normalized tables (more space efficient).

**Parameters:**

`word` - The word to look up.

**Returns:**

The category entries for the word.

---

**update**

```
public void update(java.lang.String word,  
    CategoryEntries categoryEntries)
```

---

**update**

```
public void update(java.lang.String word,  
    CategoryEntry categoryEntry)
```

---

**write**

```
public void write(java.lang.String word,  
    CategoryEntries categoryEntries)
```

---

**write**

```
public void write(java.lang.String word,  
    CategoryEntry categoryEntry)
```

---

**write3**

```
public void write3(java.lang.String word,  
    CategoryEntries categoryEntries)
```

Write a word with its category entries into the dictionary (3 tables, more space efficient).

**Parameters:**

`word` - The word to write.

`categoryEntries` - The category entries for the word.

---

**getDbType**

```
public java.lang.String getDbType()
```

---

## setDbType

```
public void setDbType(java.lang.String dbType)
```

---

---

## getDbDriver

```
public java.lang.String getDbDriver()
```

---

---

## setDbDriver

```
public void setDbDriver(java.lang.String dbDriver)
```

---

---

## getDbHost

```
public java.lang.String getDbHost()
```

---

---

## setDbHost

```
public void setDbHost(java.lang.String dbHost)
```

---

---

## getDbPort

```
public java.lang.String getDbPort()
```

---

---

## setDbPort

```
public void setDbPort(java.lang.String dbPort)
```

---

---

## getDbName

```
public java.lang.String getDbName()
```

---

---

## setDbName

```
public void setDbName(java.lang.String dbName)
```

---

---

## getDbUsername

```
public java.lang.String getDbUsername()
```

---

(continued from last page)

---

## setDbUsername

```
public void setDbUsername(java.lang.String dbUsername)
```

---

---

## getDbPassword

```
public java.lang.String getDbPassword()
```

---

---

## setDbPassword

```
public void setDbPassword(java.lang.String dbPassword)
```

---

---

## isFastMode

```
public boolean isFastMode()
```

---

---

## setFastMode

```
public void setFastMode(boolean fastMode)
```

---

---

## close

```
public void close()
```

---

---

## openReader

```
public boolean openReader()
```

---

---

## openWriter

```
public void openWriter()
```

---



# tud.iir.persistence

## Class DictionaryFileIndex

```
java.lang.Object
├── tud.iir.persistence.DictionaryIndex
│   └── tud.iir.persistence.DictionaryFileIndex
```

---

```
public class DictionaryFileIndex
extends DictionaryIndex
```

This class can be used to create, write and read a dictionary index.

**Author:**

David Urbansky

---

## Constructors

### DictionaryFileIndex

```
public DictionaryFileIndex(java.lang.String indexPath)
```

## Methods

### write

```
public void write(java.lang.String word,
    CategoryEntries categoryEntries)
```

---

### update

```
public void update(java.lang.String word,
    CategoryEntries categoryEntries)
```

---

### update

```
public void update(java.lang.String word,
    CategoryEntry categoryEntry)
```

---

### write

```
public void write(java.lang.String word,
    CategoryEntry categoryEntry)
```

---

(continued from last page)

## read

```
public CategoryEntries read(java.lang.String word)
```

---

## empty

```
public void empty()
```

---

## openWriter

```
public void openWriter()
```

---

## close

```
public void close()
```

---

## openReader

```
public boolean openReader()
```

---

## getCategories

```
public Categories getCategories()
```

---

## setCategories

```
public void setCategories(Categories categories)
```

---

# tud.iir.persistence Class DictionaryIndex

java.lang.Object

└--tud.iir.persistence.DictionaryIndex

Direct Known Subclasses:

[DictionaryDBIndexH2](#), [DictionaryDBIndexMySQL](#), [DictionaryFileIndex](#)

---

public abstract class **DictionaryIndex**  
extends java.lang.Object

---

## Constructors

### DictionaryIndex

public **DictionaryIndex**()

---

## Methods

### write

public abstract void **write**(java.lang.String word,  
[CategoryEntries](#) categoryEntries)

---

### write

public abstract void **write**(java.lang.String word,  
[CategoryEntry](#) categoryEntry)

---

### update

public abstract void **update**(java.lang.String word,  
[CategoryEntries](#) categoryEntries)

---

### update

public abstract void **update**(java.lang.String word,  
[CategoryEntry](#) categoryEntry)

---

(continued from last page)

---

## read

```
public abstract CategoryEntries read(java.lang.String word)
```

---

## empty

```
public abstract void empty()
```

---

## close

```
public abstract void close()
```

---

## openWriter

```
public abstract void openWriter()
```

---

## openReader

```
public abstract boolean openReader()
```

---

## getDictionary

```
public Dictionary getDictionary()
```

---

## setDictionary

```
public void setDictionary(Dictionary dictionary)
```

---

## setIndexPath

```
public void setIndexPath(java.lang.String indexPath)
```

---

## getIndexPath

```
public java.lang.String getIndexPath()
```

---

# tud.iir.persistence

## Class Format

```
java.lang.Object
|
|--tud.iir.persistence.Format
```

---

```
public class Format
extends java.lang.Object
```

A format for an attribute that can be specified in an xml file if xsd data types are not enough.

**Author:**  
David Urbansky

---

## Constructors

### Format

```
public Format(java.lang.String concept,
               java.lang.String attribute,
               java.lang.String description)
```

## Methods

### getConcept

```
public java.lang.String getConcept()
```

---

### setConcept

```
public void setConcept(java.lang.String concept)
```

---

### getAttribute

```
public java.lang.String getAttribute()
```

---

### setAttribute

```
public void setAttribute(java.lang.String attribute)
```

---

### getDescription

```
public java.lang.String getDescription()
```

---

---

## setDescription

```
public void setDescription(java.lang.String description)
```

# tud.iir.persistence

## Class IndexManager

```
java.lang.Object
|
|--tud.iir.persistence.IndexManager
```

---

```
public class IndexManager
extends java.lang.Object
```

Write and read from the Lucene index.

**Author:**

David Urbansky

---

## Methods

### getInstance

```
public static IndexManager getInstance()
```

---

### getIndexPath

```
public java.lang.String getIndexPath()
```

---

### writeIndex

```
public void writeIndex(java.lang.String filename,
                        java.lang.String url,
                        java.lang.String resultID)
```

---

### c

```
public void c()
    throws java.lang.Exception
```

---

### getFromIndex

```
public java.util.ArrayList getFromIndex(java.lang.String field,
                                         java.lang.String queryString)
```

---

### main

```
public static void main(java.lang.String[] args)
    throws java.lang.Exception
```

---

(continued from last page)

**Parameters:**

args



# tud.iir.persistence

## Class OntologyManager

java.lang.Object

└─ tud.iir.persistence.OntologyManager

---

public class **OntologyManager**  
extends java.lang.Object

Read and write the ontology.

**Author:**

David Urbansky, Robert Willner

---

## Methods

### getInstance

public static [OntologyManager](#) **getInstance()**

---

### loadOntology

public [KnowledgeManager](#) **loadOntology()**

Load the ontology from the standard location. Instantiate all concepts and properties for the KnowledgeManager.

---

### loadOntologyFile

public [KnowledgeManager](#) **loadOntologyFile**(java.lang.String filePath)

Load ontology from given location into the KnowledgeManager.

**Parameters:**

filePath - The file path.

---

### saveExtractions

public void **saveExtractions**([KnowledgeManager](#) knowledgeManager)

Store all extracted entities and facts into the owl knowledge base.

**Parameters:**

knowledgeManager - The knowledge manager.

---

### updateOntologyFile

public void **updateOntologyFile**([KnowledgeManager](#) knowledgeManager,  
java.io.File ontologyfile)

---

(continued from last page)

## removeConcept

```
public void removeConcept(KnowledgeManager knowledgeManager,  
    java.io.File ontologyfile,  
    int conceptId)
```

---

## removeAttribute

```
public void removeAttribute(KnowledgeManager knowledgeManager,  
    java.io.File ontologyfile,  
    int attributeId)
```

---

## jenaDBTest

```
public void jenaDBTest()
```

---

## clearCompleteKnowledgeBase

```
public void clearCompleteKnowledgeBase()
```

---

## getOntModel

```
public OntModel getOntModel(java.lang.String filePath)
```

---

## getConcepts

```
public java.util.HashSet getConcepts(OntModel om)
```

---

## getConceptProperties

```
public java.lang.String getConceptProperties(java.lang.String conceptName,  
    OntModel om)
```

---

## main

```
public static void main(java.lang.String[] args)
```

---

## tud.iir.persistence Class PersistenceManager

java.lang.Object

└─ tud.iir.persistence.PersistenceManager

---

public class **PersistenceManager**  
extends java.lang.Object

The PersistenceManager triggers the DatabaseManager and the OntologyManager.

**Author:**

David Urbansky

---

### Constructors

#### PersistenceManager

public **PersistenceManager**()

### Methods

#### saveExtractions

public static void **saveExtractions**([KnowledgeManager](#) knowledgeManager)

---

#### cleanKnowledgeBase

public static void **cleanKnowledgeBase**()

# tud.iir.persistence

## Class PredefinedSource

```
java.lang.Object
└--tud.iir.persistence.PredefinedSource
```

---

```
public class PredefinedSource
extends java.lang.Object
```

Sources can be predefined in an xml file.

**Author:**  
David Urbansky

---

## Constructors

### PredefinedSource

```
public PredefinedSource(Source source,
                        java.lang.String conceptName)
```

---

### PredefinedSource

```
public PredefinedSource(Source source,
                        java.lang.String conceptName,
                        java.util.HashSet attributeNames)
```

---

## Methods

### getSource

```
public Source getSource()
```

---

### setSource

```
public void setSource(Source source)
```

---

### getConceptName

```
public java.lang.String getConceptName()
```

---

### setConceptName

```
public void setConceptName(java.lang.String conceptName)
```

---

(continued from last page)

---

## getAttributeNames

```
public java.util.HashSet getAttributeNames()
```

---

## setAttributeNames

```
public void setAttributeNames(java.util.HashSet attributeNames)
```

---

Package  
**tud.iir.reporting**

# tud.iir.reporting

## Class ChartCreator

java.lang.Object

└─ tud.iir.reporting.ChartCreator

---

public class **ChartCreator**  
extends java.lang.Object

The ChartCreator creates charts.

**Author:**

David Urbansky

---

## Fields

### XY\_LINE\_CHART

public static final int **XY\_LINE\_CHART**

Constant value: 1

---

### XY\_SCATTER\_CHART

public static final int **XY\_SCATTER\_CHART**

Constant value: 2

---

## Constructors

### ChartCreator

public **ChartCreator**()

## Methods

### createXYChart

```
public static void createXYChart(java.lang.String fileName,  
    java.util.ArrayList dataTuplesSet,  
    java.lang.String title,  
    java.lang.String xAxis,  
    java.lang.String yAxis,  
    boolean preciseXAxis,  
    int type)
```

Create a chart, save it to the correct report folder.

---

(continued from last page)

## createVerticalBarChart

```
public static void createVerticalBarChart(java.lang.String fileName,  
    CategoryDataset data,  
    java.lang.String title,  
    java.lang.String xAxis,  
    java.lang.String yAxis)
```

Create a bar chart.

---

## createHorizontalBarChart

```
public static void createHorizontalBarChart(java.lang.String fileName,  
    CategoryDataset data,  
    java.lang.String title,  
    java.lang.String xAxis,  
    java.lang.String yAxis)
```

---

## createBarChart

```
public static void createBarChart(java.lang.String fileName,  
    CategoryDataset data,  
    java.lang.String title,  
    java.lang.String xAxis,  
    java.lang.String yAxis,  
    PlotOrientation plotOrientation)
```

---

## createLineChart

```
public static void createLineChart(java.lang.String fileName,  
    java.util.ArrayList dataTuplesSet,  
    java.util.ArrayList seriesNames,  
    java.lang.String title,  
    java.lang.String xAxis,  
    java.lang.String yAxis,  
    boolean preciseXAxis)
```

---

## main

```
public static void main(java.lang.String[] args)
```



# tud.iir.reporting Class Report

```
java.lang.Object
└--tud.iir.reporting.Report
```

```
public class Report
extends java.lang.Object
```

A Report is a list of measures and calculations. measures: totalEntities, totalCorrectEntities, entityPrecision, correctEntityPerMinute, totalFacts, totalCorrectFacts, factPrecision, correctFactsPerMinute, avgFactsPerEntity, avgCorrectFactsPerEntity, factF1 define "correct" as having a corroboration over "minEntityCorroboration" and "minFactCorroboration" as defined in the filter class

**Author:**  
David Urbansky

## Fields

### totalEntities

```
public double totalEntities
```

Values will be accessed through variables. ...ForView functions are normalized and made to print for view (in reports)

### totalCorrectEntities

```
public double totalCorrectEntities
```

### entityPrecision

```
public double entityPrecision
```

### correctEntitiesPerMinute

```
public double correctEntitiesPerMinute
```

### totalFacts

```
public double totalFacts
```

### totalCorrectFacts

```
public double totalCorrectFacts
```

---

## factPrecision

```
public double factPrecision
```

---

## correctFactsPerMinute

```
public double correctFactsPerMinute
```

---

## avgFactsPerEntity

```
public double avgFactsPerEntity
```

---

## avgCorrectFactsPerEntity

```
public double avgCorrectFactsPerEntity
```

---

## factF1

```
public double factF1
```

---

## Constructors

### Report

```
public Report()
```

## Methods

### getTotalEntitiesForView

```
public java.lang.String getTotalEntitiesForView()
```

---

### getTotalCorrectEntitiesForView

```
public java.lang.String getTotalCorrectEntitiesForView()
```

---

### getEntityPrecisionForView

```
public java.lang.String getEntityPrecisionForView()
```

---

---

## getCorrectEntitiesPerMinuteForView

```
public java.lang.String getCorrectEntitiesPerMinuteForView()
```

---

---

## getTotalFactsForView

```
public java.lang.String getTotalFactsForView()
```

---

---

## getTotalCorrectFactsForView

```
public java.lang.String getTotalCorrectFactsForView()
```

---

---

## getFactPrecisionForView

```
public java.lang.String getFactPrecisionForView()
```

---

---

## getCorrectFactsPerMinuteForView

```
public java.lang.String getCorrectFactsPerMinuteForView()
```

---

---

## getAvgFactsPerEntityForView

```
public java.lang.String getAvgFactsPerEntityForView()
```

---

---

## getAvgCorrectFactsPerEntityForView

```
public java.lang.String getAvgCorrectFactsPerEntityForView()
```

---

---

## getFactF1ForView

```
public java.lang.String getFactF1ForView()
```

---

---

## toList

```
public java.lang.String toList()
```

For saving purposes return all report values as a list.

# tud.iir.reporting Class Reporter

```
java.lang.Object
└--tud.iir.reporting.Reporter
```

---

```
public class Reporter
extends java.lang.Object
```

The Reporter creates reports.

---

## Methods

### getInstance

```
public static Reporter getInstance()
```

---

### getRuntime

```
public int getRuntime()
```

---

### setRuntime

```
public void setRuntime(int runtime)
```

---

### getReportFolderPath

```
public static java.lang.String getReportFolderPath()
```

---

### updateChartsOnly

```
public void updateChartsOnly()
```

Create a report for the current extraction process. the report will be created in the correct folder depending on whether the complete web or only a selction was used for extraction three report files will be created: 1. the complete result set with all measures for each domain will be saved 2. only the total and averaged results will be saved 3. a pdf file with a table with all measures for all domains and charts will be saved

---

### createReport

```
public void createReport(KnowledgeManager knowledgeManager)
```

---

(continued from last page)

## createDBReport

```
public void createDBReport(boolean openFile)
```

---

## main

```
public static void main(java.lang.String[] args)
```

### Parameters:

args

# tud.iir.reporting

## Class ReportFileParser

```
java.lang.Object
└--tud.iir.reporting.ReportFileParser
```

---

```
public class ReportFileParser
extends java.lang.Object
```

The ReportFileParser reads report files and builds data structures from the values that can be used to assemble reports that look back in time.

**Author:**  
David Urbansky

---

## Constructors

### ReportFileParser

```
public ReportFileParser()
```

## Methods

### getExtractionQuantities

```
public static java.util.ArrayList getExtractionQuantities()
```

Read "totalReport" files and extract data about entity and fact extractions (quantities). Automatically take the correct reporting folder.

**Returns:**

---

### getExtractionQualities

```
public static java.util.ArrayList getExtractionQualities()
```

Read "totalReport" files and extract data about entity and fact precisions and fact F1 (qualities). Automatically take the correct reporting folder.

**Returns:**

An array of values.

# tud.iir.reporting Class ReportSet

```
java.lang.Object
├-- java.util.AbstractMap
│   └-- java.util.HashMap
│       └-- tud.iir.reporting.ReportSet
```

## All Implemented Interfaces:

java.util.Map, java.io.Serializable, java.lang.Cloneable, java.util.Map

---

```
public class ReportSet
extends java.util.HashMap
```

A ReportSet holds reports (with the measures) for several domains.

## Author:

David Urbansky

---

## Constructors

### ReportSet

```
public ReportSet(int runtime)
```

---

## Methods

### getRuntime

```
public double getRuntime()
```

---

### setRuntime

```
public void setRuntime(int runtime)
```

---

### getTotalEntities

```
public double getTotalEntities()
```

Get number of all extracted entities for all domains.

## Returns:

Number of all extracted entities for all domains.

---

### getTotalEntitiesForView

```
public java.lang.String getTotalEntitiesForView()
```

---

## getTotalCorrectEntities

```
public double getTotalCorrectEntities()
```

Get number of all correct extracted entities for all domains.

**Returns:**

Number of all correct extracted entities for all domains.

---

## getTotalCorrectEntitiesForView

```
public java.lang.String getTotalCorrectEntitiesForView()
```

---

## getTotalFacts

```
public double getTotalFacts()
```

Get number of all extracted facts for all domains.

**Returns:**

Number of all extracted facts for all domains.

---

## getTotalFactsForView

```
public java.lang.String getTotalFactsForView()
```

---

## getTotalCorrectFacts

```
public double getTotalCorrectFacts()
```

Get number of all correct extracted facts for all domains.

**Returns:**

Number of all correct extracted facts for all domains.

---

## getTotalCorrectFactsForView

```
public java.lang.String getTotalCorrectFactsForView()
```

---

## getTotalEntityPrecision

```
public double getTotalEntityPrecision()
```

Get precision for all extracted entities and domains.

**Returns:**

Precision for all extracted entities and domains.

---



(continued from last page)

---

## getTotalEntityPrecisionForView

```
public java.lang.String getTotalEntityPrecisionForView()
```

---

## getTotalFactPrecision

```
public double getTotalFactPrecision()
```

Get precision for all extracted facts and domains.

**Returns:**

Precision for all extracted facts and domains.

---

## getTotalFactPrecisionForView

```
public java.lang.String getTotalFactPrecisionForView()
```

---

## getCorrectEntitiesPerMinute

```
public double getCorrectEntitiesPerMinute()
```

Get extracted correct entities per minute for all domains.

**Returns:**

Extracted correct entities per minute for all domains.

---

## getCorrectEntitiesPerMinuteForView

```
public java.lang.String getCorrectEntitiesPerMinuteForView()
```

---

## getCorrectFactsPerMinute

```
public double getCorrectFactsPerMinute()
```

Get extracted correct facts per minute for all domains.

**Returns:**

Extracted correct facts per minute for all domains.

---

## getCorrectFactsPerMinuteForView

```
public java.lang.String getCorrectFactsPerMinuteForView()
```

---

## getAvgFactsPerEntity

```
public double getAvgFactsPerEntity()
```

Get avg. precision for all extracted facts, entities and domains.

**Returns:**

---

(continued from last page)

Average precision for all extracted facts, entities and domains.

---

## getAvgFactsPerEntityForView

```
public java.lang.String getAvgFactsPerEntityForView()
```

---

## getAvgCorrectFactsPerEntity

```
public double getAvgCorrectFactsPerEntity()
```

Get avg. precision for all extracted correct facts, entities and domains.

**Returns:**

Average precision for all extracted correct facts, entities and domains.

---

## getAvgCorrectFactsPerEntityForView

```
public java.lang.String getAvgCorrectFactsPerEntityForView()
```

---

## getFactF1

```
public double getFactF1()
```

Get avg. f1 for all extracted facts and domains.

**Returns:**

Average f1 for all extracted facts and domains.

---

## getFactF1ForView

```
public java.lang.String getFactF1ForView()
```

---

## saveCompleteReportSet

```
public void saveCompleteReportSet()
```

Save the complete ReportSet in a txt file.

---

## saveTotalOnly

```
public void saveTotalOnly()
```

Save only total values (that have been calculated from all domains) in a txt file.

---

---

Package  
**tud.iir.tagging**

# tud.iir.tagging

## Class DatasetCreator

```
java.lang.Object
├--tud.iir.tagging.DatasetCreator
```

---

```
public class DatasetCreator
    extends java.lang.Object
```

The DatasetCreator crawls web pages and marks the given seed entities. The marked up pages are saved in: 1. separate (x)html files 2. separate text files (cleansed html) 3. one long text file, all text files from 2 concatenated

**Author:**  
David Urbansky

---

## Constructors

### DatasetCreator

```
public DatasetCreator(java.util.Set seedEntities)
```

## Methods

### createDataset

```
public void createDataset()
```

---

### setResultsPerEntity

```
public void setResultsPerEntity(int resultsPerEntity)
```

---

### getResultsPerEntity

```
public int getResultsPerEntity()
```

---

### getDatasetName

```
public java.lang.String getDatasetName()
```

---

### setDatasetName

```
public void setDatasetName(java.lang.String datasetName)
```

---

(continued from last page)

---

## getSeedEntities

```
public java.util.Set getSeedEntities()
```

---

## setSeedEntities

```
public void setSeedEntities(java.util.Set seedEntities)
```

---

## getDataSetLocation

```
public java.lang.String getDataSetLocation()
```

---

## setDataSetLocation

```
public void setDataSetLocation(java.lang.String dataSetLocation)
```

---

## main

```
public static void main(java.lang.String[] args)
```

### Parameters:

args

# tud.iir.tagging Class EntityList

```
java.lang.Object
├-- java.util.AbstractCollection
│   ├── java.util.AbstractList
│   │   ├── java.util.ArrayList
│   │   └-- tud.iir.tagging.EntityList
```

## All Implemented Interfaces:

java.util.Collection, java.util.List, java.io.Serializable, java.lang.Cloneable,  
java.util.RandomAccess, java.util.List

```
public class EntityList
extends java.util.ArrayList
```

## Constructors

### EntityList

```
public EntityList()
```

## Methods

### getEntity

```
public RecognizedEntity getEntity(java.lang.String name)
```

### add

```
public boolean add(RecognizedEntity e)
```

### addAll

```
public boolean addAll(java.util.Collection c)
```

## tud.iir.tagging Interface KnowledgeBaseCommunicatorInterface

All Known Implementing Classes:

[TestKnowledgeBaseCommunicator](#)

---

public interface **KnowledgeBaseCommunicatorInterface**  
extends

---

### Methods

#### categoryEntriesInKB

public [CategoryEntries](#) **categoryEntriesInKB**(java.lang.String entityName)

---

#### getTrainingEntities

public [EntityList](#) **getTrainingEntities**(double percentage)

## tud.iir.tagging Class RecognizedEntities

```
java.lang.Object
├-- java.util.AbstractCollection
│   ├── java.util.AbstractSet
│   │   ├── java.util.HashSet
│   │   └── tud.iir.tagging.RecognizedEntities
```

### All Implemented Interfaces:

java.util.Collection, java.util.Set, java.io.Serializable, java.lang.Cloneable, java.util.Set

---

```
public class RecognizedEntities
    extends java.util.HashSet
```

---

## Constructors

### RecognizedEntities

```
public RecognizedEntities()
```

## Methods

### contains

```
public boolean contains(java.lang.Object obj)
```

Check whether ArrayList contains obj.

#### Returns:

True if the obj is contained, false otherwise.



## tud.iir.tagging

# Class RecognizedEntity

java.lang.Object

└--tud.iir.tagging.RecognizedEntity

---

public class **RecognizedEntity**  
extends java.lang.Object

---

## Constructors

### RecognizedEntity

```
public RecognizedEntity(java.lang.String name,  
                        CategoryEntries categories,  
                        double trust)
```

## Methods

### getName

```
public java.lang.String getName()
```

---

### setName

```
public void setName(java.lang.String name)
```

---

### hasCategoryEntries

```
public boolean hasCategoryEntries()
```

---

### getCategoryEntries

```
public CategoryEntries getCategoryEntries()
```

---

### setCategoryEntries

```
public void setCategoryEntries(CategoryEntries categories)
```

---

(continued from last page)

## addCategoryEntry

```
public void addCategoryEntry(CategoryEntry categoryEntry)
```

---

## addCategoryEntries

```
public void addCategoryEntries(CategoryEntries categoryEntries)
```

---

## getTrust

```
public double getTrust()
```

---

## setTrust

```
public void setTrust(double trust)
```

---

## addTrust

```
public void addTrust(double trust)
```

---

## equals

```
public boolean equals(java.lang.Object obj)
```

---

## toString

```
public java.lang.String toString()
```

---

# tud.iir.tagging

## Class StringTagger

java.lang.Object

└--tud.iir.tagging.StringTagger

---

public class **StringTagger**  
extends java.lang.Object

---

## Constructors

### StringTagger

public **StringTagger**()

## Methods

### tagAndSaveString

public static void **tagAndSaveString**(java.io.File input)

---

### tagString

public static java.lang.String **tagString**(java.io.File f)

---

### tagString

public static java.lang.String **tagString**(java.lang.String s)

---

### getTaggedEntities

public static [Annotations](#) **getTaggedEntities**(java.lang.String text)

---

### main

public static void **main**(java.lang.String[] args)

---

# tud.iir.tagging

## Class TestKnowledgeBaseCommunicator

java.lang.Object

└─ tud.iir.tagging.TestKnowledgeBaseCommunicator

All Implemented Interfaces:

[KnowledgeBaseCommunicatorInterface](#)

---

```
public class TestKnowledgeBaseCommunicator
    extends java.lang.Object
    implements KnowledgeBaseCommunicatorInterface
```

---

## Constructors

### TestKnowledgeBaseCommunicator

```
public TestKnowledgeBaseCommunicator()
```

## Methods

### categoryEntriesInKB

```
public CategoryEntries categoryEntriesInKB(java.lang.String entityName)
```

---

### getTrainingEntities

```
public EntityList getTrainingEntities(double percentage)
```

---

Package

**tud.iir.visualization.applets**

# tud.iir.visualization.applets

## Class PrefuseGraph

```
java.lang.Object
  |
  +--JPrefuseApplet
       |
       +--tud.iir.visualization.applets.PrefuseGraph
```

---

```
public class PrefuseGraph
  extends JPrefuseApplet
```

---

## Constructors

### PrefuseGraph

```
public PrefuseGraph()
```

## Methods

### init

```
public void init()
```

---

### createGraph

```
public javax.swing.JComponent createGraph(java.lang.String datafile,
                                           java.lang.String label)
```

---

### showGraph

```
public javax.swing.JComponent showGraph(Graph graph,
                                          int focusNodeID)
```

---

---

Package  
**tud.iir.web**

# tud.iir.web

## Class AggregatedResult

```
java.lang.Object
└-- tud.iir.web.AggregatedResult
```

All Implemented Interfaces:  
java.io.Serializable

```
public class AggregatedResult
    extends java.lang.Object
    implements java.io.Serializable
```

The knowledge unit aggregated result. The AggregatedResults generated by the SourceAggregator contain references to the WebResults they result from, as well as an aggregated rank value. They are not stored in the database directly, but keep a reference to the Source to which they refer.

**Author:**  
Christopher Friedrich

## Constructors

### AggregatedResult

```
public AggregatedResult(java.util.List webresults,
                        float aggregatedRank)
```

## Methods

### getWebresults

```
public java.util.List getWebresults()
```

### getAggregatedRank

```
public float getAggregatedRank()
```

### getSource

```
public Source getSource()
```

### getSearchEngines

```
public java.util.Set getSearchEngines()
```



# tud.iir.web

## Class ConnectionTimeout

java.lang.Object

└─ tud.iir.web.ConnectionTimeout

### All Implemented Interfaces:

java.lang.Runnable

---

public final class **ConnectionTimeout**  
extends java.lang.Object  
implements java.lang.Runnable

The ConnectionTimeout is necessary because java does not set timeouts when a server starts sending data and stops without sending an end signal.

### Author:

David Urbansky, Philipp Katz

---

## Constructors

### ConnectionTimeout

```
public ConnectionTimeout(java.net.URLConnection urlConnection,  
                          int timeout)
```

## Methods

### run

```
public final void run()
```

---

### isActive

```
public boolean isActive()
```

---

### setActive

```
public void setActive(boolean active)
```

# tud.iir.web Class Crawler

```
java.lang.Object
└--tud.iir.web.Crawler
```

```
public class Crawler
extends java.lang.Object
```

The Crawler downloads pages from the web. List of proxies can be found here: <http://www.proxy-list.org/en/index.php> TODO handle namespace in xpath TODO some methods here are duplicates from [PageAnalyzer](#) or could be moved there: [extractBodyContent\(Document\)](#), [extractBodyContent\(String, boolean\)](#), [extractDescription\(Document\)](#), [extractKeywords\(Document\)](#), [extractTitle\(Document\)](#)

**Author:**

David Urbansky, Philipp Katz, Martin Werner

## Fields

### DEFAULT\_CONNECTION\_TIMEOUT

```
public static final int DEFAULT_CONNECTION_TIMEOUT
```

the default connection timeout  
Constant value: 10000

### DEFAULT\_READ\_TIMEOUT

```
public static final int DEFAULT_READ_TIMEOUT
```

the default read timeout when retrieving pages  
Constant value: 16000

### DEFAULT\_OVERALL\_TIMEOUT

```
public static final int DEFAULT_OVERALL_TIMEOUT
```

the default overall timeout (after which the connection is reset)  
Constant value: 60000

### DEFAULT\_NUM\_RETRIES

```
public static final int DEFAULT_NUM_RETRIES
```

the default number of retries when downloading fails.  
Constant value: 0

### BYTES

```
public static final int BYTES
```

Constant value: 1

(continued from last page)

## KILO\_BYTES

```
public static final int KILO_BYTES
```

Constant value: 2

## MEGA\_BYTES

```
public static final int MEGA_BYTES
```

Constant value: 3

## GIGA\_BYTES

```
public static final int GIGA_BYTES
```

Constant value: 4

## sessionDownloadedBytes

```
public static long sessionDownloadedBytes
```

keep track of the total number of bytes downloaded by all crawler instances used

## Constructors

### Crawler

```
public Crawler()
```

### Crawler

```
public Crawler(int connectionTimeout,  
               int readTimeout,  
               int overallTimeout)
```

### Crawler

```
public Crawler(java.lang.String configPath)
```

## Methods

### loadConfig

```
public final void loadConfig(java.lang.String configPath)
```

Load the configuration file from the specified location and set the variables accordingly.

#### Parameters:

`configPath` - The location of the configuration file.

---

## startCrawl

```
public void startCrawl(java.util.HashSet urlStack,  
    boolean inDomain,  
    boolean outDomain)
```

---

## startCrawl

```
public void startCrawl(java.lang.String startURL,  
    boolean inDomain,  
    boolean outDomain)
```

---

## setStopCount

```
public void setStopCount(int number)
```

---

## addOnlyFollow

```
public void addOnlyFollow(java.lang.String follow)
```

---

## addURLRule

```
public void addURLRule(java.lang.String rule)
```

---

## saveURLDump

```
public void saveURLDump(java.lang.String filename)
```

---

## getLinks

```
public java.util.HashSet getLinks(boolean inDomain,  
    boolean outDomain)
```

Get a set of links from the source page.

### Parameters:

`inDomain` - If true all links that point to other pages within the same domain of the source page are added.

`outDomain` - If true all links that point to other pages outside the domain of the source page are added.

### Returns:

A set of urls.

---

(continued from last page)

---

## getLinks

```
public java.util.HashSet getLinks(boolean inDomain,  
    boolean outDomain,  
    java.lang.String prefix)
```

---

## getLinks

```
public java.util.HashSet getLinks(org.w3c.dom.Document document,  
    boolean inDomain,  
    boolean outDomain)
```

---

## getLinks

```
public java.util.HashSet getLinks(org.w3c.dom.Document document,  
    boolean inDomain,  
    boolean outDomain,  
    java.lang.String prefix)
```

---

## getDomain

```
public static java.lang.String getDomain(java.lang.String url,  
    boolean includeProtocol)
```

Return the root/domain URL. For example: `http://www.example.com/page.html` is converted to `http://www.example.com`

### Parameters:

`url`

`includeProtocol` - include protocol prefix, e.g. "`http://`"

### Returns:

root URL, or empty String if URL cannot be determined, never `null`

---

## getDomain

```
public static java.lang.String getDomain(java.lang.String url)
```

Return the root/domain URL. For example: `http://www.example.com/page.html` is converted to `http://www.example.com`

### Parameters:

`url`

### Returns:

root URL, or empty String if URL cannot be determined, never `null`

---

## extractTitle

```
public static java.lang.String extractTitle(org.w3c.dom.Document webPage)
```

(continued from last page)

---

## extractBodyContent

```
public static java.lang.String extractBodyContent(org.w3c.dom.Document webPage)
```

---

## extractBodyContent

```
public static java.lang.String extractBodyContent(java.lang.String pageContent,  
boolean textOnly)
```

Extracts the content of the body out of a given pageContent; textOnly-Parameter allows to get the textual content

---

## extractKeywords

```
public static java.util.ArrayList extractKeywords(org.w3c.dom.Document webPage)
```

---

## extractDescription

```
public static java.util.ArrayList extractDescription(org.w3c.dom.Document webPage)
```

---

## makeFullURL

```
public static java.lang.String makeFullURL(java.lang.String pageUrl,  
java.lang.String baseUrl,  
java.lang.String linkUrl)
```

Creates a full/absolute URL based on the specified parameters. Handling links in HTML documents can be tricky. If no absolute URL is specified in the link itself, there are two factors for which we have to take care:

1. The document's URL
2. If provided, a base URL inside the document, which can be as well be absolute or relative to the document's URL

### Parameters:

pageUrl - actual URL of the document.

baseUrl - base URL defined in document's header, can be `null` if no base URL is specified.

linkUrl - link URL from the document to be made absolute.

### Returns:

the absolute URL, empty String, if URL cannot be created or for mailto and javascript links, never `null`.

### See Also:

[HTML base & Basis-Adresse einer Webseite](#)

---

## makeFullURL

```
public static java.lang.String makeFullURL(java.lang.String pageUrl,  
java.lang.String linkUrl)
```

---

---

## getSiblingPage

```
public java.lang.String getSiblingPage(java.lang.String url)
```

---

---

## getSiblingPage

```
public java.lang.String getSiblingPage(org.w3c.dom.Document document)
```

---

---

## getCleanURL

```
public static java.lang.String getCleanURL(java.lang.String url)
```

---

---

## removeAnchors

```
public static java.lang.String removeAnchors(java.lang.String url)
```

---

---

## getUserAgent

```
public java.lang.String getUserAgent()
```

---

---

## setDocument

```
public void setDocument(org.w3c.dom.Document document)
```

---

---

## setDocument

```
public void setDocument(java.lang.String url)
```

---

---

## setDocument

```
public void setDocument(java.lang.String url,  
                        boolean isXML,  
                        boolean callback)
```

---

---

## getDocument

```
public org.w3c.dom.Document getDocument()
```

---

(continued from last page)

---

## getWebDocument

```
public org.w3c.dom.Document getWebDocument(java.lang.String url)
```

Get a web page ((X)HTML document).

**Parameters:**

`url` - The URL or file path of the web page.

**Returns:**

The W3C document.

---

## getWebDocument

```
public org.w3c.dom.Document getWebDocument(java.lang.String url,  
boolean callback)
```

Get a web page ((X)HTML document).

**Parameters:**

`url` - The URL or file path of the web page.

`callback` - set to `false` to disable callback for this document.

**Returns:**

The W3C document.

---

## getXMLDocument

```
public org.w3c.dom.Document getXMLDocument(java.lang.String url)
```

Get XML document from a URL. Pure XML documents can be created with the native DocumentBuilderFactory, which works better with the native XPath queries.

**Parameters:**

`url` - The URL or file path pointing to the XML document.

**Returns:**

The XML document.

---

## getXMLDocument

```
public org.w3c.dom.Document getXMLDocument(java.lang.String url,  
boolean callback)
```

Get XML document from a URL. Pure XML documents can be created with the native DocumentBuilderFactory, which works better with the native XPath queries.

**Parameters:**

`url` - The URL or file path pointing to the XML document.

`callback` - set to `false` to disable callback for this document.

**Returns:**

The XML document.

---

## getJSONDocument

```
public JSONObject getJSONDocument(java.lang.String url)
```

Get a json object from a URL. The retrieved contents must return a valid json object.

---



(continued from last page)

**Parameters:**

url - The url pointing to the json string.

**Returns:**

The json object.

---

## download

```
public java.lang.String download(java.lang.String urlString,  
    boolean stripTags,  
    boolean stripComments,  
    boolean stripJSAndCSS,  
    boolean joinTagsAndRemoveNewlines)
```

Download the contents that are retrieved from the given URL.

**Parameters:**

urlString - The URL of the desired contents.

stripTags - If true, HTML tags will be stripped (but not comments, js and css tags).

stripComments - If true, comment tags will be stripped.

stripJSAndCSS - If true, JavaScript and CSS tags will be stripped

joinTagsAndRemoveNewlines - If true, multiple blank spaces and line breaks will be removed.

**Returns:**

The contents as a string.

---

## download

```
public java.lang.String download(java.lang.String urlString)
```

---

## download

```
public java.lang.String download(java.lang.String urlString,  
    boolean stripTags)
```

---

## downloadNotBlacklisted

```
public java.lang.String downloadNotBlacklisted(java.lang.String urlString)
```

Only download if the urlString is in a valid form and the file-ending is not blacklisted (see Extractor.java for file-ending-blackList)

---

## downloadAndSave

```
public void downloadAndSave(java.util.HashSet urlSet)
```

---

## downloadAndSave

```
public void downloadAndSave(java.io.File file)
```

---

## downloadAndSave

```
public void downloadAndSave(java.io.File file,  
    int startLine)
```

---

## downloadAndSave

```
public boolean downloadAndSave(java.lang.String urlString,  
    java.lang.String path)
```

---

## downloadImage

```
public void downloadImage(java.lang.String url,  
    java.lang.String path)
```

---

## getTotalDownloadSize

```
public double getTotalDownloadSize()
```

---

## getTotalDownloadSize

```
public double getTotalDownloadSize(int unit)
```

---

## setTotalDownloadSize

```
public void setTotalDownloadSize(int totalDownloadSize)
```

---

## getLastDownloadSize

```
public int getLastDownloadSize()
```

---

## setLastDownloadSize

```
public void setLastDownloadSize(int lastDownloadSize)
```

---

## getSessionDownloadSize

```
public static double getSessionDownloadSize(int unit)
```

---

## getCrawlerCallbacks

```
public java.util.Set getCrawlerCallbacks()
```

---

---

## addCrawlerCallback

```
public void addCrawlerCallback(CrawlerCallback crawlerCallback)
```

---

---

## removeCrawlerCallback

```
public void removeCrawlerCallback(CrawlerCallback crawlerCallback)
```

---

---

## getMaxThreads

```
public int getMaxThreads()
```

---

---

## setMaxThreads

```
public void setMaxThreads(int maxThreads)
```

---

---

## getThreadCount

```
public int getThreadCount()
```

---

---

## increaseThreadCount

```
public void increaseThreadCount()
```

---

---

## decreaseThreadCount

```
public void decreaseThreadCount()
```

---

---

## getProxy

```
public java.net.Proxy getProxy()
```

Returns the current Proxy.

**Returns:**

---

(continued from last page)

---

## setProxy

```
public void setProxy(java.net.Proxy proxy)
```

Sets the current Proxy.

### Parameters:

proxy

---

## setSwitchProxyRequests

```
public void setSwitchProxyRequests(int switchProxyRequests)
```

Number of requests after the proxy is changed.

### Parameters:

switchProxyRequests - number of requests for proxy change. Must be greater than 1 or -1 which means: change never.

---

## getSwitchProxyRequests

```
public int getSwitchProxyRequests()
```

---

## addToProxyList

```
public void addToProxyList(java.lang.String proxyEntry)
```

Add an entry to the proxy list. The entry must be formatted as "HOST:PORT".

### Parameters:

proxyEntry - The proxy to add.

---

## setProxyList

```
public void setProxyList(java.util.List proxyList)
```

Set a list of proxies. Each entry must be formatted as "HOST:PORT".

### Parameters:

proxyList - The list of proxies.

---

## getProxyList

```
public java.util.List getProxyList()
```

---

## changeProxy

```
public void changeProxy()
```

Cycle the proxies, taking the first item from the queue and adding it to the end.

---

## checkProxy

```
public boolean checkProxy()
```

---

(continued from last page)

Check whether the currently set proxy is working.

**Returns:**

True if proxy returns result, false otherwise.

---

## getHeaders

```
public java.util.Map getHeaders(java.lang.String pageURL)
```

Get HTTP Headers of an URLConnection to pageURL.

---

## isValidURL

```
public static boolean isValidURL(java.lang.String url,  
    boolean checkHTTPResp)
```

Check if an URL is in a valid form and the file-ending is not blacklisted (see Extractor.java for blacklist) TODO: remove checkHTTPRespParameter

**Parameters:**

url - the URL

checkHTTPResp - the check http resp

**Returns:**

true, if is a valid URL

---

## downloadBinaryFile

```
public static java.io.File downloadBinaryFile(java.lang.String urlString,  
    java.lang.String pathWithFileName)
```

Download a binary file from specified URL to a given path.

**Parameters:**

urlString - the urlString

pathWithFileName - the path where the file should be saved

**Returns:**

the file

---

## downloadInputStream

```
public java.io.InputStream downloadInputStream(java.net.URL url)  
    throws java.io.IOException
```

Download from specified URL. This method caches the incoming InputStream and blocks until all incoming data has been read or the timeout has been reached.

**Parameters:**

url - The URL to download.

**Returns:**

The input stream.

**Throws:**

IOException

(continued from last page)

---

## downloadInputStream

```
public java.io.InputStream downloadInputStream(java.lang.String urlString)
    throws java.io.IOException
```

Download from specified URL string. This method caches the incoming InputStream and blocks until all incoming data has been read or the timeout has been reached.

**Parameters:**

urlString

**Returns:****Throws:**

IOException

---

## getResponseCode

```
public int getResponseCode(java.lang.String urlString)
```

Get the response code of the given url after sending a HEAD request. This works only for HTTP connections.

**Parameters:**

urlString - The URL.

**Returns:**

The HTTP response Code.

---

## verifyURL

```
public static java.lang.String verifyURL(java.lang.String urlCandidate,
    java.lang.String pageURL)
```

Check URL for validness and eventually modify e.g. relative path

**Parameters:**

urlCandidate - the URLCandidate

pageURL - the pageURL

**Returns:**

the verified URL

---

## setUseCompression

```
public void setUseCompression(boolean useCompression)
```

Use to disable compression.

**Parameters:**

useCompression - If true, compression will be used, if false then not.

---

## setConnectionTimeout

```
public void setConnectionTimeout(int connectionTimeout)
```

---

## getConnectionTimeout

```
public int getConnectionTimeout()
```

---

---

## setReadTimeout

```
public void setReadTimeout(int readTimeout)
```

---

---

## getReadTimeout

```
public int getReadTimeout()
```

---

---

## setOverallTimeout

```
public void setOverallTimeout(int overallTimeout)
```

---

---

## getOverallTimeout

```
public int getOverallTimeout()
```

---

---

## isFeedAutodiscovery

```
public boolean isFeedAutodiscovery()
```

---

---

## setFeedAutodiscovery

```
public void setFeedAutodiscovery(boolean feedAutodiscovery)
```

---

---

## setNumRetries

```
public void setNumRetries(int numRetries)
```

---

---

## getNumRetries

```
public int getNumRetries()
```

---

---

## documentToString

```
public static java.lang.String documentToString(org.w3c.dom.Document document)
```

---

(continued from last page)

Get the string representation of a document.

**Parameters:**

`document` - The document.

**Returns:**

The string representation of the document.

---

## main

```
public static void main(java.lang.String[] args)
```

**Parameters:**

`args`



## tud.iir.web Interface CrawlerCallback

All Known Implementing Classes:

[FeedDiscoveryCallback](#), [ObjectExtractor](#)

---

public interface **CrawlerCallback**  
extends

An interface for the CrawlerCallback.

**Author:**

David Urbansky

---

### Methods

#### **crawlerCallback**

```
public void crawlerCallback(org.w3c.dom.Document document)
```

# tud.iir.web Class FeedFinder

```
java.lang.Object
|
+--tud.iir.web.FeedFinder
```

---

**Deprecated.** *@see tud.iir.news.NewsAggregator instead*

```
public class FeedFinder
extends java.lang.Object
```

The FeedFinder downloads links to feeds on the web and stores them in the database.

**Author:**  
David Urbansky

---

## Constructors

### FeedFinder

```
public FeedFinder()
```

**Deprecated.**

## Methods

### searchFeeds

```
public static void searchFeeds()
```

**Deprecated.**

---

### main

```
public static void main(java.lang.String[] args)
```

**Deprecated.**

# tud.iir.web

## Class RankAggregation

```
java.lang.Object
├-- tud.iir.web.RankAggregation
```

---

```
public class RankAggregation
extends java.lang.Object
```

RankAggregation combines multiple ranked lists of WebResults into one. This class is described in detail in "Friedrich, Christopher. WebSnippets - Extracting and Ranking of entity-centric knowledge from the Web. Diploma thesis, Technische Universität Dresden, April 2010".

**Author:**  
Christopher Friedrich

---

## Fields

### RANK\_AVERAGE

```
public static final int RANK_AVERAGE
```

Constant value: 0

## Constructors

### RankAggregation

```
public RankAggregation()
```

## Methods

### aggregate

```
public static java.util.List aggregate(java.util.List lists,
                                     int method,
                                     int maxResults)
```

The interface to access different rank aggregation techniques.

**Parameters:**

`lists` - - List of ranked lists of WebResults  
`method` - - The technique to use for rank aggregation. Currently implemented is `RANK_AVERAGE`.  
`maxResults` - - The maximum number of results returned in the resulting, aggregated list.

**Returns:**

A list of AggregatedResult's

# tud.iir.web

## Class SourceAggregator

```
java.lang.Object
└--tud.iir.web.SourceAggregator
```

---

```
public class SourceAggregator
extends java.lang.Object
```

A collection of source aggregation algorithms. All algorithms take an entity as input and return an list of AggregatedResults as output, given the provided aggregation technique and rank aggregation technique. This class is described in detail in "Friedrich, Christopher. WebSnippets - Extracting and Ranking of entity-centric knowledge from the Web. Diploma thesis, Technische Universität Dresden, April 2010".

**Author:**

Christopher Friedrich

---

## Fields

### IFM

```
public static final int IFM
```

Constant value: 0

## Constructors

### SourceAggregator

```
public SourceAggregator()
```

## Methods

### getIndices

```
public int[] getIndices(Entity currentEntity)
```

### aggregateWebResults

```
public java.util.List aggregateWebResults(Entity currentEntity,
    int method,
    int maxResults,
    int rankAggregationMethod)
```

The main function to access the algorithms implemented in this class. It takes an entity as input and provides a list of AggregatedResults as output.

**Parameters:**

currentEntity - - The entity to retrieve AggregatedResults for

---

(continued from last page)

`method` - - The source aggregation technique to use

`maxResults` - - The maximum length of the results list

`rankAggregationMethod` - - the rank aggregation method to use

#### Returns:

The list of `AggregatedResults`

---

## aggregateWebResults

```
public java.util.List aggregateWebResults(Entity currentEntity,  
    int method,  
    int[] indices,  
    int maxResults,  
    int rankAggregationMethod)
```

---

## main

```
public static void main(java.lang.String[] abc)
```

# tud.iir.web

## Class SourceRetriever

```
java.lang.Object
└--tud.iir.web.SourceRetriever
```

---

```
public class SourceRetriever
extends java.lang.Object
```

The SourceRetriever queries the indices of Yahoo!, Google, Microsoft and Hakia.

**Author:**  
David Urbansky, Christopher Friedrich, Philipp Katz

---

## Fields

### LANGUAGE\_ENGLISH

```
public static final int LANGUAGE_ENGLISH
```

Constant value: 0

---

### LANGUAGE\_GERMAN

```
public static final int LANGUAGE_GERMAN
```

Constant value: 1

---

## Constructors

### SourceRetriever

```
public SourceRetriever()
```

## Methods

### getResultCount

```
public int getResultCount()
```

---

### setResultCount

```
public void setResultCount(int resultCount)
```

---

(continued from last page)

---

## getSource

```
public int getSource()
```

---

## setSource

```
public void setSource(int source)
```

---

## getImages

```
public final java.util.ArrayList getImages(java.lang.String searchQuery,  
    int source,  
    boolean exact,  
    java.lang.String[] matchContent)
```

Get a list of images for a given query.

### Parameters:

`searchQuery` - The query.

`source` - The code of the source.

`exact` - If true, the query must match exactly, otherwise it is a sequence of terms.

`matchContent` - All match content keywords must appear in the caption of the image.

### Returns:

A list of images.

---

## getHitCount

```
public final int getHitCount(java.lang.String searchQuery)
```

Return number of hits for a given query.

### Parameters:

`searchQuery` - A search query.

### Returns:

The number of hits for a given query.

---

## getURLs

```
public java.util.ArrayList getURLs(java.lang.String searchQuery,  
    boolean exact)
```

---

## getURLs

```
public java.util.ArrayList getURLs(java.lang.String searchQuery)
```

---

## getURLs

```
public java.util.ArrayList getURLs(java.lang.String searchQuery,  
    int source)
```

---

## getURLs

```
public java.util.ArrayList getURLs(java.lang.String searchQuery,  
    int source,  
    boolean exact)
```

---

## getWebResults

```
public final java.util.ArrayList getWebResults(java.lang.String searchQuery,  
    int source,  
    boolean exact)
```

Returns a list of WebResults for a search engine query.

### Parameters:

searchQuery - - The search query string to use  
source - - Which search engine to query  
exact - - Whether to put search terms in quotes

### Returns:

---

## getWebResultsFromGoogle

```
public java.util.ArrayList getWebResultsFromGoogle(java.lang.String searchQuery,  
    java.lang.String languageCode)
```

---

## getLanguage

```
public int getLanguage()
```

---

## setLanguage

```
public void setLanguage(int language)
```

---

## main

```
public static void main(java.lang.String[] args)
```



# tud.iir.web

## Class SourceRetrieverManager

```
java.lang.Object
└--tud.iir.web.SourceRetrieverManager
```

```
public class SourceRetrieverManager
extends java.lang.Object
```

The SourceRetrieverManager holds information about query settings and statistics for indices of Yahoo!, Google, Microsoft, Hakia, Bing, Twitter and Google Blogs. The SourceRetrieverManager is singleton.

**Author:**  
David Urbansky, Christopher Friedrich, Philipp Katz

## Fields

### YAHOO

```
public static final int YAHOO
```

Constant value: 1

### GOOGLE

```
public static final int GOOGLE
```

Constant value: 2

### MICROSOFT

```
public static final int MICROSOFT
```

Constant value: 3

### HAKIA

```
public static final int HAKIA
```

Constant value: 4

### YAHOO\_BOSS

```
public static final int YAHOO_BOSS
```

Constant value: 5

### BING

```
public static final int BING
```

(continued from last page)

Constant value: 6

---

## TWITTER

```
public static final int TWITTER
```

Constant value: 7

---

## GOOGLE\_BLOGS

```
public static final int GOOGLE_BLOGS
```

Constant value: 8

---

## TEXTRUNNER

```
public static final int TEXTRUNNER
```

Constant value: 9

---

## YAHOO\_BOSS\_NEWS

```
public static final int YAHOO_BOSS_NEWS
```

Constant value: 10

---

## Methods

### getInstance

```
public static SourceRetrieverManager getInstance()
```

---

### getResultCount

```
public int getResultCount()
```

---

### setResultCount

```
public void setResultCount(int resultCount)
```

---

### getSource

```
public int getSource()
```

---

(continued from last page)

---

## setSource

```
public void setSource(int source)
```

---

## getRequestCount

```
public int getRequestCount(int source)
```

Get total number of requests that have been made to the given source.

**Parameters:**

`source` - The code for the source.

**Returns:**

The number of requests for the given source or -1 if the source code was invalid.

---

## addRequest

```
public void addRequest(int source)
```

Count a request for a source.

**Parameters:**

`source` - The code for the source.

---

## getSearchEngines

```
public static int[] getSearchEngines()
```

Get all indices of search engines available.

**Returns:**

An array of indices.

---

## getLogs

```
public java.lang.String getLogs()
```

Get a log string of how many request have been sent.

**Returns:**

A log string.

---

## getName

```
public static java.lang.String getName(int source)
```

Get a human readable string for search engine constant.

**Parameters:**

`source`

**Returns:**

name of the corresponding search engine.

---

(continued from last page)

## main

```
public static void main(java.lang.String[] args)
```

## tud.iir.web Class URLDownloader

```
java.lang.Object
└--tud.iir.web.URLDownloader
```

---

```
public class URLDownloader
extends java.lang.Object
```

Allows simultaneous downloading of multiple URLs. The resulting InputStreams are cached by this class and can be processed after all downloads are done. TODO merge this into Crawler.

**Author:**  
Philipp Katz

---

### Constructors

#### URLDownloader

```
public URLDownloader()
```

### Methods

#### start

```
public void start(URLDownloader.URLDownloaderCallback callback)
```

---

#### start

```
public void start()
```

---

#### add

```
public void add(java.lang.String urlString)
```

---

#### get

```
public java.io.InputStream get(java.lang.String urlString)
```

---

#### getAll

```
public java.util.Collection getAll()
```

---

---

## setMaxThreads

```
public void setMaxThreads(int maxThreads)
```

---

## getMaxThreads

```
public int getMaxThreads()
```

---

## setMaxFails

```
public void setMaxFails(int maxFails)
```

---

## getMaxFails

```
public int getMaxFails()
```

---

## main

```
public static void main(java.lang.String[] args)  
    throws java.net.MalformedURLException
```

## tud.iir.web Interface URLDownloader.URLDownloaderCallback

---

public interface URLDownloader.URLDownloaderCallback  
extends

---

### Methods

#### finished

```
public void finished(java.lang.String url,  
                    java.io.InputStream inputStream)
```

# tud.iir.web

## Class URLRankingCache

java.lang.Object

└─ tud.iir.web.URLRankingCache

---

```
public class URLRankingCache
    extends java.lang.Object
```

Cache for [URLRankingServices](#). As those APIs have a considerable latency, we cache their results for a specific time in the database. TODO caching ttl sometimes does not work correctly.

**Author:**  
Philipp Katz

---

## Constructors

### URLRankingCache

```
public URLRankingCache()
```

## Methods

### getSource

```
public Source getSource(java.lang.String url)
```

Get a Source object for the specified url. Return `null` if no such Source.

**Parameters:**  
`url`

**Returns:**

---

### get

```
public java.util.Map get(Source source)
```

Get cached ranking values for specified Source. Returns only those values which are under the specified TTL or an empty list if there are no cached or up-to-date ranking values, never `null`.

**Parameters:**  
`source`

**Returns:**

---

### add

```
public void add(Source source,
               java.util.Map rankings)
```



(continued from last page)

Adds or updates rankings for a specific Source in the cache.

**Parameters:**

source  
rankings

---

## setTtlSeconds

```
public void setTtlSeconds(int ttlSeconds)
```

Set the TTL for the cache. Set to -1 to never expire the cached data.

**Parameters:**

ttlSeconds

---

## main

```
public static void main(java.lang.String[] args)
```

# tud.iir.web

## Class URLRankingServices

```
java.lang.Object
└--tud.iir.web.URLRankingServices
```

---

```
public class URLRankingServices
extends java.lang.Object
```

This class provides access to external, Web 2.0 typical services with APIs which offer ranking indicators for web pages. Some of them are taken from "SEO for Firefox" extension. API key are configured in "config/apikeys.conf". <http://tools.seobook.com/firefox/seo-for-firefox.html> TODO possibility to disable caching TODO specific caching for domains

**Author:**  
Philipp Katz

---

## Constructors

### URLRankingServices

```
public URLRankingServices()
```

## Methods

### setServices

```
public void setServices(java.util.Collection services)
```

Define the services which to check. Use this, if you do not want to check all available services. For instance, if you only want to check Google Page Rank and Yahoo! Page links, use:

```
setServices(Arrays.asList(new Service[] { Service.GOOGLE_PAGE_RANK,
Service.YAHOO_PAGE_LINKS }));
```

**Parameters:**

services

---

### getRanking

```
public java.util.Map getRanking(java.lang.String url)
```

Get ranking for supplied url from all specified ranking services. By default, all available services are checked, see [URLRankingServices.Service.values\(\)](#). Use [setServices\(Collection\)](#) to specify the services to be checked by this method.

**Parameters:**

url

**Returns:**

---

(continued from last page)

## getRanking

```
public java.util.Map getRanking(Source source)
```

Get ranking for supplied Source from all specified ranking services. By default, all available services are checked, see [URLRankingServices.Service.values\(\)](#). Use [setServices\(Collection\)](#) to specify the services to be checked by this method.

### Parameters:

url

### Returns:

---

## getRanking

```
public float getRanking(Source source,  
    URLRankingServices.Service service)
```

Retrieve the ranking for a specific url from a specific service. Results are *not* cached.

### Parameters:

url

service

### Returns:

---

## main

```
public static void main(java.lang.String[] args)  
    throws java.lang.Exception
```

---

## setCacheTtlSeconds

```
public void setCacheTtlSeconds(int ttlSeconds)
```

### Parameters:

ttlSeconds

### See Also:

[URLRankingCache.setTtlSeconds\(int\)](#)

# tud.iir.web

## Class URLRankingServices.Service

```
java.lang.Object
  |
  +- java.lang.Enum
        +- tud.iir.web.URLRankingServices.Service
```

### All Implemented Interfaces:

java.io.Serializable, java.lang.Comparable

```
public static final class URLRankingServices.Service
extends java.lang.Enum
```

Type safe enum for all available ranking services.

## Fields

### BITLY\_CLICKS

```
public static final tud.iir.web.URLRankingServices.Service BITLY_CLICKS
```

Get the number of clicks for the specified URL on bit.ly. This is now the default URL shortening service on Twitter, so this measure is a good indicator for the popularity of this URL on microblogging platforms.

### DIGGS

```
public static final tud.iir.web.URLRankingServices.Service DIGGS
```

Get the number of diggs for the specified URL. If there are multiple entries for the URL, sum up all diggs.

### MIXX\_VOTES

```
public static final tud.iir.web.URLRankingServices.Service MIXX_VOTES
```

Get the number of Mixx votes. Mixx is a mix of social networking and bookmarking platform.

### REDDIT\_SCORE

```
public static final tud.iir.web.URLRankingServices.Service REDDIT_SCORE
```

Get the reddit score. This is determined by the number of up/down votes on the reddit site.

### DELICIOUS\_POSTS

```
public static final tud.iir.web.URLRankingServices.Service DELICIOUS_POSTS
```

Get the number of posts on social bookmarking platform Delicious.

### YAHOO\_DOMAIN\_LINKS

```
public static final tud.iir.web.URLRankingServices.Service YAHOO_DOMAIN_LINKS
```

(continued from last page)

Get the number of results from Yahoo! pointing to the URL's domain.

---

## YAHOO\_PAGE\_LINKS

```
public static final tud.iir.web.URLRankingServices.Service YAHOO_PAGE_LINKS
```

Get the number of results from Yahoo! pointing to the URL.

---

## TWEETS

```
public static final tud.iir.web.URLRankingServices.Service TWEETS
```

Get the number of Tweets containing the URL's domain. It makes no sense to search for full page links as they are too long for Twitter in most cases. Use [BITLY\\_CLICKS](#) as an indicator instead.

---

## GOOGLE\_PAGE\_RANK

```
public static final tud.iir.web.URLRankingServices.Service GOOGLE_PAGE_RANK
```

Retrieves the PageRank for specified URL.

---

## GOOGLE\_DOMAIN\_PAGE\_RANK

```
public static final tud.iir.web.URLRankingServices.Service GOOGLE_DOMAIN_PAGE_RANK
```

Retrieve the PageRank for URL's domain from Google.

---

## ALEXA\_RANK

```
public static final tud.iir.web.URLRankingServices.Service ALEXA_RANK
```

Get Alexa popularity rank.

---

## MAJESTIC\_SEO

```
public static final tud.iir.web.URLRankingServices.Service MAJESTIC_SEO
```

Get number of referring domains for specified URL from Majestic-SEO.

---

## COMPETE\_RANK

```
public static final tud.iir.web.URLRankingServices.Service COMPETE_RANK
```

Get "Domain ranking based on Unique Visitor estimate for month/year" from Compete.

## Methods

### values

```
public static URLRankingServices.Service\[\] values()
```

---

### valueOf

```
public static URLRankingServices.Service valueOf(java.lang.String name)
```

## tud.iir.web Class URLStack

```
java.lang.Object
  |-- java.util.AbstractCollection
        |-- java.util.AbstractSet
              |-- java.util.HashSet
                    |-- tud.iir.web.URLStack
```

### All Implemented Interfaces:

java.util.Collection, java.util.Set, java.io.Serializable, java.lang.Cloneable, java.util.Set

---

```
public class URLStack
extends java.util.HashSet
```

Stack of URLs. TODO replace with native Java stack?

### Author:

David Urbansky

---

## Constructors

### URLStack

```
public URLStack()
```

## Methods

### contains

```
public boolean contains(java.lang.Object o)
```

# tud.iir.web

## Class WebResult

```
java.lang.Object
├-- tud.iir.web.WebResult
```

---

```
public class WebResult
    extends java.lang.Object
```

The knowledge unit web result. WebResults are retrieved by the SourceRetriever and represent web search results.

**Author:**  
Christopher Friedrich

---

## Constructors

### WebResult

```
public WebResult(int index,
                  int rank,
                  Source source,
                  java.lang.String title,
                  java.lang.String summary)
```

## Methods

### getIndex

```
public int getIndex()
```

---

### getRank

```
public int getRank()
```

---

### getTitle

```
public java.lang.String getTitle()
```

---

### getSummary

```
public java.lang.String getSummary()
```

---

(continued from last page)

## getSource

```
public Source getSource()
```

---

## getUrl

```
public java.lang.String getUrl()
```

---

## toString

```
public java.lang.String toString()
```



---

Package

**tud.iir.web.apiwrapper**

# tud.iir.web.apiwrapper

## Class WSW

```
java.lang.Object
├-- tud.iir.web.apiwrapper.WSW
```

---

```
public class WSW
extends java.lang.Object
```

---

## Constructors

### WSW

```
public WSW(java.lang.String wswPath)
```

## Methods

### getWebServiceIDs

```
public java.util.ArrayList getWebServiceIDs(int profileID)
```

---

### createQueryURL

```
public java.lang.String createQueryURL(int webServiceID,
    java.util.HashSet parameterBindings)
```

---

### callProfile

```
public OutputObject callProfile(int profileID,
    java.util.HashSet parameterBindings)
    throws JSONException
```

---

### callWebService

```
public OutputObject callWebService(int webServiceID,
    java.util.HashSet parameterBindings)
    throws JSONException
```

---

### main

```
public static void main(java.lang.String[] args)
```

---

(continued from last page)

**Parameters:**

args

---

Package

**tud.iir.web.datasetcrawler**

# tud.iir.web.datasetcrawler

## Class DeliciousCrawler

```
java.lang.Object
└--tud.iir.web.datasetcrawler.DeliciousCrawler
```

---

```
public class DeliciousCrawler
extends java.lang.Object
```

The DeliciousCrawler creates a data set of web pages with delicious tags. This data set can then be used as training data for the web page classifier.

**Author:**  
David Urbansky

---

## Constructors

### DeliciousCrawler

```
public DeliciousCrawler()
```

## Methods

### crawl

```
public void crawl()
```

---

### cleanDataSet

```
public static void cleanDataSet(int minAppearance)
```

Read the data set, clean it and write the output to a new file.

**Parameters:**

`minAppearance` - Number of times a tag must appear in order to keep it.

---

### analyzeDataSet

```
public static void analyzeDataSet(java.lang.String suffix)
```

---

### normalizeTag

```
public static java.lang.String normalizeTag(java.lang.String tag)
```

Normalize vocabulary. For example, blogs => blogs / musica, musik => music / e-learning, learning => learn

**Parameters:**

`tag` - The tag that should be normalized.

---

(continued from last page)

**Returns:**

The normalized tag.

---

**main**

```
public static void main(java.lang.String[] args)
```

**Parameters:**

args

# tud.iir.web.datasetcrawler

## Class LanguageDatasetCompiler

java.lang.Object

└--tud.iir.web.datasetcrawler.LanguageDatasetCompiler

---

public class **LanguageDatasetCompiler**  
extends java.lang.Object

This class compiles a training set of web pages with certain languages. This training set can then be used to learn a language classifier.

**Author:**

David Urbansky

---

## Constructors

### LanguageDatasetCompiler

public **LanguageDatasetCompiler**()

## Methods

### compileDataset

public void **compileDataset**(int pagesPerLanguage)

Compiles a dataset for learning a classifier. It processes the following steps: 1. Query Google for each language to obtain web pages in the given language 2. Download x web pages and generate an entry in the dataset file. 3. Save the dataset file.

**Parameters:**

pagesPerLanguage - Number of pages per language.

---

### main

public static void **main**(java.lang.String[] args)

**Parameters:**

args

# tud.iir.web.datasetcrawler

## Class LinkedDataStatisticsCrawler

java.lang.Object

└─tud.iir.web.datasetcrawler.LinkedDataStatisticsCrawler

---

public class **LinkedDataStatisticsCrawler**  
extends java.lang.Object

The LinkedDataStatisticsCrawler creates a [GraphML](#) representation of the linked data on the web.

**Author:**

David Urbansky

---

## Constructors

### LinkedDataStatisticsCrawler

public **LinkedDataStatisticsCrawler**()

## Methods

### createGraphML

public void **createGraphML**()

Create the graph XML.

---

### main

public static void **main**(java.lang.String[] args)

**Parameters:**

args



# tud.iir.web.datasetcrawler

## Class QuoteCrawler

java.lang.Object

└─ tud.iir.web.datasetcrawler.QuoteCrawler

---

public class **QuoteCrawler**  
extends java.lang.Object

Crawl quotes.

**Author:**

David Urbansky

---

## Constructors

### QuoteCrawler

public **QuoteCrawler**()

## Methods

### extractQuotes

public void **extractQuotes**()

Extracts quotes

---

### main

public static void **main**(java.lang.String[] args)

**Parameters:**

args

# Index

## A

- AAUML 560
- AbstractMIOTypeExtractor 337
- add 3, 9, 34, 242, 702, 741, 744
- addAbsoluteRelevance 12
- addAll 4, 9, 702
- addAllowedFiletype 67
- addAnswer 568
- addAssessmentInstance 657
- addAttribute 541
- addAttributeSynonym 656
- addCategoryEntries 706
- addCategoryEntry 99, 705
- addConcept 87, 564
- addConcepts 564
- addCrawlerCallback 723
- addDataset 151
- addDocument 508, 512, 516, 518, 520, 522, 525
- addDocumentsFromFile 508
- addDocumentSimilarity 513, 516, 518, 520, 522, 525
- addDuplicate 587
- addEntity 543
- addEntry 242
- addExtraction 228, 257
- addExtractionByType 228
- addFact 657, 658
- addFactAndValue 545
- addFactForBenchmark 545
- addFactValue 552
- addFeed 614, 631, 634, 640
- addFeedEntry 615, 632, 634
- addFeeds 640
- addFeedsFromFile 641
- addFile 508
- addFileHeader 457
- addFromFile 33
- addIgnore 620
- addNamespaceToXPath 501
- addNode 493
- addNumber 335
- addOnlyFollow 716
- addPredefinedSource 532
- addQA 396
- addQAs 656
- addQueries 619
- addQuery 618
- addQuestionHash 402
- addRangeNodeDummies 533
- addRangeValue 533, 537
- addRanking 586
- addRequest 739
- addSnippets 545
- addSource 550, 556
- addSources 550
- addStats 197
- addSuffixesToBlackList 231
- addSynonym 530, 540
- addTestResult 48
- addToDictionary 107
- addToProxyList 724
- addToVocabulary 42, 43
- addToVocabularyFromFile 43
- addTrust 706
- addURLRule 716
- addURLToStack 400
- addWord 166
- aggregate 313, 641, 731
- aggregateContinuously 641
- AggregatedResult 712
- aggregateEvents 318
- aggregateWebResults 732, 733
- AlchemyNER 294
- ALEXA\_RANK 749
- allKeywords 563
- AMP 559
- analyzeContentAndSetFeatures 384
- analyzeDataSet 757
- analyzeMIOPages 387
- analyzeSWFContent 346
- Annotation 270
- Annotations 273
- AnswerClassifier 158
- AnswerFeatures 160
- appendFile 454
- appendToFile 454
- appendToFileName 451

- AppletExtractor 338
- applyExtractionTemplate 389
- arguments 466, 630
- arrangeByDate 431
- arrangeMapByDate 431, 432
- ArrayHelper 422
- ASCENDING 423
- assignCategoryEntries 99
- assignTags 616
- Attribute 529
- AttributeRange 536
- AUML 560
- AverageClassifierPerformance 132
- avgCorrectFactsPerEntity 690
- avgFactsPerEntity 690
- B**
  - badWords 346
  - BAYES\_NET 13
  - bayesRelevance 11
  - BENCHMARK\_ENTITY\_EXTRACTION 222
  - BENCHMARK\_FACT\_EXTRACTION 222
  - BENCHMARK\_FULL\_SET 221
  - BENCHMARK\_HALF\_SET 221
  - BING 737
  - BITLY\_CLICKS 748
  - BitPermutations 506
  - BODY\_CONTENT\_KEYWORDS 562
  - BodyDate 199
  - BooleanEntityTrustVoting 75
  - BRACKETS 280
  - bracketToColumn 275
  - bracketToXML 275
  - breakLineLoop 466
  - BYTES 714
- C**
  - c 679
  - calcContDateAttr 181
  - calcContDateContent 182
  - calcStringRelevance 371
  - calculateAllCharNGrams 491
  - calculateAllWordNGrams 491
  - calculateAttributeSynonyms 567
  - calculateAttributeSynonymTrust 657
  - calculateCategoryPriors 22
  - calculateCharNGrams 490
  - calculateDedicatedPageTrust 339
  - calculateldf 114
  - calculateListSimilarity 471, 472
  - calculatePrior 6
  - calculatePriors 4
  - calculateRelativeRelevances 9
  - calculateRMSE 471
  - calculateSimilarity 482
  - calculateSplit 69
  - calculateWordNGrams 490
  - callback 64
  - callProfile 754
  - callWebService 754
  - camelCaseToWords 484, 485
  - capitalizedWordCount 480
  - car 345
  - Categories 3
  - categories 122
  - Category 5
  - CategoryEntries 8
  - categoryEntriesInKB 703, 708
  - CategoryEntry 11
  - changeCheckApproach 614
  - changeProxy 724
  - CHAR\_NGRAMS 152
  - ChartCreator 687
  - CHECK\_ADAPTIVE 598
  - CHECK\_FIXED 598
  - CHECK\_PROBABILISTIC 598
  - checkDayMonthYearOrder 177, 181
  - checkForDate 187
  - checkLinkSet 197
  - checkProxy 724
  - checkTextnode 189
  - checkURLs 197
  - chunk 320
  - CLASS\_CHUNKED 609
  - CLASS\_CONSTANT 609
  - CLASS\_ON\_THE\_FLY 610
  - CLASS\_SLICED 609
  - CLASS\_SPONTANEOUS 609

CLASS\_UNKNOWN 609  
CLASS\_ZOMBIE 609  
ClassificationDocument 97  
ClassificationDocuments 101  
ClassificationTypeSetting 134  
ClassificationTypeTagSetting 136  
Classifier 13  
ClassifierEntityTrustVoting 76  
ClassifierManager 103  
ClassifierPerformance 138  
classify 86, 88, 89, 91, 94, 108, 111, 125, 169, 321, 580, 610  
classifyBinary 17  
classifySoft 17, 87  
classifyTestDocuments 108, 124  
cleanDataSet 757  
cleanDirectory 456  
cleanKnowledgeBase 683  
cleanUnusedOntologyElements 662  
clear 446  
clearCompleteDatabase 662  
clearCompleteKnowledgeBase 682  
clearEntities 542  
clearFeedTables 616  
clearRangeValues 533, 537  
cloneDocument 638  
close 667, 672, 674, 676  
closeIndexWriter 20  
CollectionHelper 423  
COLON\_FACT\_REPRESENTATION 575  
COLON\_PHRASE 227  
color 385  
COLUMN 280  
columnBIOToColumn 275  
columnToBracket 274  
columnToColumnBIO 274  
columnToXML 274  
columnTrainingToTest 275  
COMBINED\_TRUST 221  
CombinedClassifier 105  
compare 73, 219, 252, 261, 332, 355, 438, 439, 588  
COMPETE\_RANK 749  
compileDataset 759  
concat 422  
concatMatchedString 483  
Concept 538  
ConceptDateComparator 219  
ConnectionTimeout 713  
constructAllXPath 234  
constructXPath 235  
contains 3, 34, 406, 424, 584, 704, 750  
containsCategoryName 3  
containsDataObject 428  
containsDate 443  
containsMIO 341  
containsNumber 477  
containsProperNoun 477  
containsSearchWordOrMorphs 380  
ContentDate 202  
ControlledTagger 42  
ControlledTaggerEvaluation 46  
ControlledTaggerEvaluationResult 48  
ControlledTaggerEvaluationSettings 50  
ControlledTaggerSettings 55, 56  
convert 176  
convertNodeToString 503  
copyDirectory 456  
copyFile 455  
CORRECT 285  
correctEntitiesPerMinute 689  
correctFactsPerMinute 690  
CORRECTNESS\_MARGIN 551  
countAll 196  
countDates 432  
countEntityOccurrences 410  
Counter 426  
CountMap 427  
countOccurrences 637  
countSame 196  
countTagLength 461  
countTags 461  
countThreads 196  
countWhitespaces 486  
countWords 482  
crawl 757  
Crawler 715  
crawlerCallback 389, 622, 729  
crawlURLwithDate 197  
createActualDate 193

- createBarChart 688
- createBenchmarkConcepts 566
- createBenchmarkIndex 256
- createDataset 700
- createDBReport 692
- createDirectory 459
- createDocument 638
- createEntityFile 79
- createEntityFile2 79
- createEntityQuery 417
- createEntityTrustChart 79
- createFactLog 328, 329
- createFocusedCrawlQuery 260
- createGraph 710
- createGraphML 760
- createGUI 419
- createHorizontalBarChart 688
- createLineChart 688
- createPhraseQuery 260
- createQueryURL 754
- createReport 692
- createSeedQuery 260
- createSnippetBenchmarks 566
- createTemplate 389
- createVerticalBarChart 687
- createXYChart 687
- CROSS\_TRUST 221
- crossValidate 146
- CrossValidationResult 143
- CrossValidator 146
  
- D**
  
- DataHolder 428
- Dataset 148
- DatasetCallback 64
- DatasetCreator 599, 700
- DatasetEntry 65
- DatasetFilter 67
- DATE0 570
- DATE1 570
- DATE2 570
- DATE3 570
- DATE4 571
- DATE\_ANSI\_C 575
- DATE\_ANSI\_C\_TZ 575
- DATE\_BODY\_STRUC 562
- DATE\_EU\_D\_MM 573
- DATE\_EU\_D\_MM\_Y 573
- DATE\_EU\_D\_MM\_Y\_T 573
- DATE\_EU\_D\_MMMM 573
- DATE\_EU\_D\_MMMM\_Y 573
- DATE\_EU\_D\_MMMM\_Y\_T 573
- DATE\_EU\_MM\_Y 573
- DATE\_EUSA\_MMMM\_Y 575
- DATE\_ISO8601\_YD 572
- DATE\_ISO8601\_YD\_NO 572
- DATE\_ISO8601\_YD\_T 571
- DATE\_ISO8601\_YM 571
- DATE\_ISO8601\_YMD 571
- DATE\_ISO8601\_YMD\_NO 572
- DATE\_ISO8601\_YMD\_SEPARATOR 571
- DATE\_ISO8601\_YMD\_SEPARATOR\_T 571
- DATE\_ISO8601\_YMD\_T 571
- DATE\_ISO8601\_YW 571
- DATE\_ISO8601\_YW\_NO 572
- DATE\_ISO8601\_YWD 571
- DATE\_ISO8601\_YWD\_NO 572
- DATE\_ISO8601\_YWD\_T 571
- DATE\_RFC\_1036 575
- DATE\_RFC\_1036\_UTC 575
- DATE\_RFC\_1123 575
- DATE\_RFC\_1123\_UTC 575
- DATE\_URL 572
- DATE\_URL\_D 572
- DATE\_URL\_MMMM\_D 572
- DATE\_URL\_SPLIT 573
- DATE\_USA\_MM\_D 574
- DATE\_USA\_MM\_D\_Y 573
- DATE\_USA\_MM\_D\_Y\_SEPARATOR 574
- DATE\_USA\_MM\_D\_Y\_SEPARATOR\_1 574
- DATE\_USA\_MM\_D\_Y\_SEPARATOR\_2 574
- DATE\_USA\_MM\_D\_Y\_SEPARATOR\_3 574
- DATE\_USA\_MM\_D\_Y\_T 574
- DATE\_USA\_MM\_Y 574
- DATE\_USA\_MMMM\_D 575
- DATE\_USA\_MMMM\_D\_Y 574
- DATE\_USA\_MMMM\_D\_Y\_T 574
- DateArrayHelper 430
- DateComparator 438

DateConverter 176  
DateEvaluator 177  
DateEvaluatorHelper 179  
DateGetter 183  
DateGetterMain 191  
DateHelper 443  
DateNormalizer 645  
DATEPOS\_IN\_DOC 201  
DATEPOS\_IN\_TAGTEXT 201  
DAY 205  
DAY\_MS 442  
DB\_H2 20  
DB\_INDEX\_FAST 19  
DB\_INDEX\_NORMALIZED 19  
DB\_MYSQL 19  
DBStore 446  
decodeBase64 484  
decreaseFrequency 5  
decreaseThreadCount 230, 723  
decrement 426  
DedicatedPageDetector 339  
DEEP\_CORRELATIONS 60  
DEFAULT\_CONNECTION\_TIMEOUT 714  
DEFAULT\_CORRELATION\_WEIGHT 55  
DEFAULT\_N\_GRAM\_LENGTH 507  
DEFAULT\_NUM\_RETRIES 714  
DEFAULT\_OVERALL\_TIMEOUT 714  
DEFAULT\_PRIOR\_WEIGHT 55  
DEFAULT\_READ\_TIMEOUT 714  
DEFAULT\_SIMILARITY\_THRESHOLD 507  
DEFAULT\_SKETCH\_SIZE 507  
DEFAULT\_TAG\_COUNT 55  
DEFAULT\_TAG\_MATCH\_PATTERN 55  
DEFAULT\_TFIDF\_THRESHOLD 55  
delete 456  
deleteFeedEntryById 616  
deleteIndex 511, 515, 517, 519, 520, 523, 525  
DELICIOUS\_POSTS 748  
DeliciousCrawler 757  
DeliciousDatasetReader 62  
DeliciousDatasetSplitter 69  
demo 296, 302, 303, 304  
deployMetaDates 177  
DESCENDING 423  
deserialize 455  
detectAnswer 396  
detectFactTable 234  
detectRolePages 377  
determineFeedTextType 612  
Dictionary 20  
DictionaryClassifier 107  
DictionaryDBIndexH2 664  
DictionaryDBIndexMySQL 669  
DictionaryFileIndex 673  
DictionaryIndex 675  
DIFFG 591  
DIGGS 748  
discoverEntityXPath 263  
discoverFeeds 619  
DISTANCE\_DATE\_KEYWORD 201  
DOCUMENT\_SENTENCES 409  
DOCUMENT\_SNIPPETS 409  
documentToString 727  
doesTrainedMIOClassifierExists 95  
done 385  
download 721  
downloadAndSave 594, 721, 722  
downloadBinaryFile 725  
downloadFeed 642  
downloadImage 722  
downloadInputStream 725  
downloadNotBlacklisted 721  
  
E  
  
empty 664, 669, 674, 676  
emptyElement 250  
emptyIndex 20  
emptyWhitsp 559  
encodeBase64 484  
endElement 250  
englishStopWords 152  
enterTextnodes 189  
ENTITY 570  
Entity 544  
ENTITY\_FOCUSED\_CRAWL 227  
ENTITY\_PHRASE 227  
ENTITY\_SEED 227  
EntityAssessor 77  
EntityClassifier 78

EntityDateComparator 252  
entityExtractionIsRunning 220  
EntityExtractionProcess 253  
EntityExtractionThread 254  
EntityFactExtractionThread 323  
EntityList 702  
EntityMIOExtractionThread 340  
entityPrecision 689  
EntitySnippetExtractionThread 408  
EntityTrustComparator 261  
EntityTrustVoting 79  
entriesUniform 263  
equals 7, 35, 37, 72, 583, 706  
ERROR1 284  
ERROR2 285  
ERROR3 285  
ERROR4 285  
ERROR5 285  
escapeForRegularExpression 478  
evaluate 46, 88, 89, 177, 278  
evaluateBenchmarkExtractions 566  
evaluateBenchmarkExtractionsGetPAR 566  
evaluateKeyLocAttr 180  
evaluateKeyLocCont 181  
evaluateNER 299, 305  
evaluateTag 179  
evaluateURLDate 178  
evaluateURLwithDate 197  
EvaluationAnnotation 283  
EvaluationHelper 82  
EvaluationResult 285  
EvaluationSetting 149  
Event 309  
EventAggregator 313  
EventAggregatorException 315  
EventFeatureExtractor 318  
EXACT\_MATCH 284  
extract 82, 256, 265, 267  
Extractable 548  
extractBodyContent 717, 718  
extractDescription 718  
ExtractedDate 205, 206  
ExtractedDateHelper 192  
ExtractedImage 586  
ExtractedImageComparator 588  
extractEventFromURL 316  
extractFacts 329, 333  
extractFactsForEntityName 328  
extractFAQ 395  
EXTRACTION\_SOURCES 173  
EXTRACTION\_TYPE\_TRUST 221  
extractionFocusedCrawl 255  
extractionFromPhrase 255  
ExtractionProcessManager 222  
extractionSeeds 255  
ExtractionType 227  
extractKeywords 718  
Extractor 229  
extractPOSFromSentence 415  
extractQuotes 761  
extractSnippets 409  
extractTagElement 464  
extractTitle 717  
extractXMLContent 356  
  
F  
  
Fact 551  
FactExtractionDecisionTree 324  
factExtractionIsRunning 220  
FactExtractionProcess 326  
factF1 690  
factPrecision 690  
FactString 330  
FactValue 555  
FactValueComparator 332  
faculty 470  
FAQ 398  
FastMIODetector 341  
FastWordCorrelationMatrix 26  
FeatureEntityTrustVoting 83  
FeatureEvaluator 27  
FeatureObject 29  
FeatureSetting 152  
Feed 601  
FeedClassifier 610  
FeedContentClassifier 612  
FeedDiscovery 618  
FeedEntry 623  
FeedFinder 730

FeedPostStatistics 627  
FeedProcessingAction 630  
FeedStoreDummy 634  
file 196  
fileContentToLines 452  
fileExists 459  
FileFormatParser 274  
FileHelper 450  
fillDomainsForFactExtractionTest 566  
filter 430, 431  
FILTER\_FULL\_DATE 430  
FILTER\_IS\_IN\_RANGE 429  
FILTER\_KEYLOC\_ATTR 430  
FILTER\_KEYLOC\_CONT 430  
FILTER\_KEYLOC\_NO 430  
FILTER\_TECH\_ARCHIVE 430  
FILTER\_TECH\_HTML\_CONT 429  
FILTER\_TECH\_HTML\_HEAD 429  
FILTER\_TECH\_HTML\_STRUC 429  
FILTER\_TECH\_HTTP\_HEADER 429  
FILTER\_TECH\_REFERENCE 430  
FILTER\_TECH\_URL 429  
filterAnswerCandidates 396  
filterFormat 431  
filterURLs 230  
findALLDates 188  
findDate 187  
findEntityColumn 263  
findEntityConnection 79  
findFeeds 619  
findLastBoxSection 236  
findNodeKeyword 189  
findNodeKeywordPart 189  
findPaginationURLs 262  
finished 743  
finishTest 47, 70  
finishTrain 47, 70  
FIRST\_PRIORITY 562  
firstPriorityKeywords 563  
FIXED\_COUNT 59  
FlashExtractor 342  
font 385  
Format 677  
FORMAT\_ATOM 600  
FORMAT\_RSS 600

FREE\_TEXT\_SENTENCE 226

FullPageClassifier 110

## G

generateSearchQueries 369  
gerundToInfinitive 497  
get 23, 200, 202, 207, 214, 427, 446, 741, 744  
get1PartRegExp 576  
get2Digits 193  
get2PartRegExp 576  
get3PartRegExp 576  
get4DigitYear 193  
getAbsoluteCorrelation 38  
getAbsoluteRelevance 12  
getAccuracyForCategory 140  
getAdded 602, 624  
getAddedSQLTimestamp 602, 625  
getAggregatedRank 712  
getAggregatedResult 580  
getAll 207, 741  
getAllAnswersXPath 400  
getAllRegExp 576  
getAltText 354  
getAnnotations 276, 277, 294, 296, 298, 301, 303, 305, 306  
getAnnotationsFromColumn 276  
getAnnotationsFromXMLFile 276  
getAnnotationsFromXMLText 276  
getAnswerFeatures 396  
getAnswerPrefix 400  
getAnswers 568  
getAnswerSuffix 400  
getAnswerWordCount 160  
getArrayAsString 483  
getAsFeatureObject 160  
getAssignedCategoryEntries 99  
getAssignedCategoryEntriesByRelevance 99  
getAssignedCategoryEntryNames 99  
getAssignedTags 66  
getAssignments 286  
getAttribute 324, 542, 552, 677  
getAttributeNames 541, 685  
getAttributeRanges 532  
getAttributeRangesToDelete 533



getAttributes 541  
getAttributesAsList 541  
getAttributesToDelete 541  
getAverageAccuracy 142  
getAverageF 141  
getAverageGray 594  
getAveragePerformanceDataSetTrainingFolds 144  
getAveragePerformanceFolds 145  
getAveragePerformanceTrainingFolds 144  
getAveragePrecision 141  
getAverageRecall 141  
getAverageSensitivity 141  
getAverageSpecificity 141  
getAverageTagOccurence 53  
getAvgCorrectFactsPerEntity 698  
getAvgCorrectFactsPerEntityForView 691, 698  
getAvgFactsPerEntity 697  
getAvgFactsPerEntityForView 691, 698  
getAvgFOne 48  
getAvgPrecision 48  
getAvgRecall 48  
getAvgTagCount 48  
getBadWords 347  
getBenchmarkPMIs 654  
getBenchmarkSet 225  
getBenchmarkSetSize 224  
getBenchmarkType 225  
getBestAnswerXPath 399  
getBlackList 231  
getBodyStructureDates 186  
getByKey 446  
getCategories 23, 122, 138, 674  
getCategory 11  
getCategoryByName 4  
getCategoryEntries 12, 22, 705  
getCategoryEntry 8  
getCheckApproach 607  
getCheckApproachName 607  
getCheckInterval 608  
getChecks 603  
getChildNode 503  
getChildNodes 503  
getChildren 494  
getChildrenDates 186  
getChosenClassifier 16  
getChosenClassifierName 16  
getClassAssociation 30  
getClassAssociationAsString 30  
getClassificationType 126, 135, 138  
getClassificationTypeSetting 126, 143  
getClassificationTypeTagSetting 135  
getClassifiedAs 100  
getClassifiedAsReadable 100  
getClassifiedNumberOfCategory 101  
getClassifier 16, 143  
getClassifierFeatureCombination 27  
getClassName 610  
getClassType 6, 24  
getCleanURL 719  
getCompareDepth 440  
getConcept 531, 544, 565, 677  
getConceptID 373, 659  
getConceptName 684  
getConceptProperties 682  
getConcepts 256, 565, 682  
getConcreteTags 463  
getConfig 173  
getConnection 652  
getConnectionTimeout 727  
getContentAsString 363  
getContentDates 189  
getCorrectEntitiesPerMinute 697  
getCorrectEntitiesPerMinuteForView 691, 697  
getCorrectFactsPerMinute 697  
getCorrectFactsPerMinuteForView 691, 697  
getCorrectlyAssignedCategoryEntries 121  
getCorrectValue 553  
getCorrelation 26, 40  
getCorrelations 26, 40  
getCorrelationType 56  
getCorrelationWeight 57  
getCorroboration 553, 556  
getCorroboration1 557  
getCorroboration2 557  
getCorroboration3 557  
getCorroboration4 557  
getCorroboration5 557  
getCount 373, 426  
getCountDown 473  
getCountOfXPath 242

getCrawler 393  
getCrawlerCallbacks 723  
getCurrentDatetime 443, 444  
getCurrentSource 323  
getDatabaseType 25  
getDataObject 428  
getDataSetLocation 701  
getDatasetName 700  
getDatasets 151  
getDate 183  
getDateFromString 188  
getDateString 206  
getDatetime 443  
getDbDriver 447, 666, 671  
getDbHost 447, 666, 671  
getDbName 667, 671  
getDbPassword 448, 667, 672  
getDbPort 448, 666, 671  
getDbType 447, 666, 670  
getDbUsername 448, 667, 671  
getDedicatedPageTrust 364  
getDescendants 494  
getDescription 677  
getDictionary 109, 676  
getDifference 440  
getDifferentDatesMap 435  
getDirectURL 351  
getDistinctTagCount 163  
getDocument 719  
getDocumentAsString 233  
getDocumentsForHash 512, 516, 518, 521, 522, 526  
getDocumentsForSketch 512, 515, 516, 522  
getDocumentTextDump 233, 234  
getDocumentType 100  
getDomain 717  
getDuplicateCount 586  
getElapsedTime 474  
getElapsedTimeString 474  
getEndIndex 271  
getEntities 542  
getEntitiesByDate 542  
getEntitiesByTrust 542  
getEntitiesForExtractionType 655  
getEntitiesForSource 655  
getEntity 271, 324, 351, 416, 543, 579, 702  
getEntityChunks 310, 410  
getEntityFeatures 310  
getEntityIDsByName 657  
getEntityName 333, 657  
getEntityPrecisionForView 690  
getEntityQuery 265  
getEntries 603  
getEntryText 625  
getEntryURL 399  
getEqualDate 440  
getEvaluation 17  
getEvaluationResult 47  
getEvaluationSetting 146  
getEventmap 313  
getEvents 313  
getExactestDates 435  
getExactestMap 436  
getExactness 208  
getExtractedAt 549, 557  
getExtractedAtAsUTCString 549  
getExtractionCount 546  
getExtractionLimit 257  
getExtractionQualities 694  
getExtractionQuantities 694  
getExtractions 256, 266  
getExtractionStatusDownloadedBytes 655  
getExtractionType 330, 582  
getExtractionTypeCount 546  
getExtractionTypes 260, 546, 556  
getExtractionTypesForSource 656  
getF1 132, 286  
getF1For 286  
getFact 555  
getFactF1 698  
getFactF1ForView 691, 698  
getFactForAttribute 545  
getFactPrecisionForView 691  
getFacts 545  
getFactString 330  
getFactStrings 325  
getFactValue 553, 581  
getFactValueForValue 552  
getFeature 30, 353, 579, 625  
getFeatureCombination 17

getFeatureNames 30  
getFeatures 29, 310, 353, 579, 625  
getFeatureSetting 126, 144  
getFeedById 615, 632, 635  
getFeedByUrl 615, 631, 634  
getFeedEntries 615, 616, 632, 635  
getFeedEntriesForEvaluation 616  
getFeedEntryById 615  
getFeedEntryByRawId 615, 632, 634, 635  
getFeedEntryIdsTaggedAs 616, 633, 635  
getFeedId 623  
getFeedPostDistribution 614  
getFeedProcessingAction 607  
getFeeds 615, 619, 631, 634  
getFeedUrl 601  
getFForCategory 139  
getFile 66  
getFileName 352, 451  
getFilePath 451  
getFiles 457  
getFileSize 353  
getFileType 451  
getFiletype 65  
getFindPageURL 351  
getFirstElement 436  
getFirstRealCategory 98  
getFirstTableCell 237  
getFirstWords 637  
getFormat 207, 602  
getFrequency 5, 113  
getFromIndex 679  
getFullEntityName 167  
getFullPath 495  
getFvWekaAttributes 14  
getGrayDifference 594  
getGreenPrefix 401  
getGreenUrlDepth 401  
getHeadDates 187  
getHeaders 725  
getHEADRegExp 577  
getHeight 589  
getHighestCountXPath 242, 243  
getHighestRate 179, 436  
getHitCount 735  
getHostname 363, 372  
getHow 312  
getHTMLSymbols 560  
getHTMLText 240  
getHTMLTextByXPath 239  
getHTTPHeaderDate 186  
getHTTPRegExp 577  
getID 373, 538, 548, 551, 582  
getId 601, 623  
getIdf 114  
getIdfCount 53  
getIdfIndex 52  
getIframeMioPages 344  
getIframeParentPage 365  
getIframeParentPageTitle 365  
getImageContent 590  
getImages 735  
getIncTimeRegExp 576  
getIndex 44, 114, 751  
getIndexName 511, 514, 523  
getIndexPath 24, 676, 679  
getIndexType 24  
getIndices 732  
getInnerXml 638  
getInstance 173, 232, 255, 260, 316, 328, 347,  
359, 389, 395, 414, 417, 419, 428, 606, 614, 622,  
652, 679, 681, 692, 738  
getInteractivityGrade 351  
getJSONDocument 720  
getK 111  
getKbCommunicator 307  
getKeyByValue 424  
getKeyLocToString 202  
getKeyword 208, 213  
getKeywordPriority 181  
getkFolds 149  
getKnowledgeManager 229, 540  
getLabel 493  
getLanguage 602, 736  
getLastChecked 605  
getLastDownloadSize 722  
getLastFeedEntry 604  
getLastFeedEntrySQLTimestamp 604  
getLastHeadlines 604  
getLastInsertID 659  
getLastSearched 540, 549

getLeafPath 495  
getLength 271  
getLengthSim 639  
getLevenshteinSim 639  
getLink 624  
getLinkedMioPages 349  
getLinkName 363  
getLinkParentPage 363  
getLinks 716, 717  
getLinkTitle 364  
getLogger 230, 256  
getLoggerName 469  
getLogs 739  
getLongestCommonString 482  
getLongestGap 471  
getLongestHighCountXPath 243  
getLongestPostGap 628  
getMainCategoryEntry 98  
getMainContent 583  
getMatchingImageURL 591  
getMatchingImageURLs 592  
getMaxCheckInterval 603  
getMaxFails 742  
getMaximumTermLength 153  
getMaximumURLs 399  
getMaxNGramLength 153  
getMaxTags 136  
getMaxTerms 153  
getMaxThreads 314, 723, 742  
getMeanSquareError 594  
getMedianDifference 471  
getMedianPostGap 628  
getMessage 469  
getMeticulousPostDistribution 604  
getMinCheckInterval 603  
getMinimumTermLength 154  
getMinkowskiSimilarity 594  
getMinN 510  
getMinNGramLength 153  
getMinTags 136  
getMIOType 352  
getMIOTypes 347  
getMonthNumber 192  
getMostLikelyCategoryEntry 9, 21, 22  
getMostLikelyTag 272  
getMostLikelyTagName 272  
getN 113  
getName 5, 23, 71, 125, 279, 398, 539, 549, 705, 739  
getNewName 532, 539  
getNewSuperClass 539  
getNewSynonyms 533, 540  
getNextSibling 236, 237  
getNextTableCell 237  
getNextTableRow 238  
getNGram 116  
getnGramLength 509  
getNode 494, 502  
getNodeByID 503  
getNodes 502  
getNormalizedDate 206  
getNormalizedNumber 649  
getNumberOfCorrectClassifiedDocumentsInCategory 139  
getNumberOfDocuments 22, 116, 512, 517, 519, 521, 523, 526  
getNumberOfExtractions 545  
getNumberOfLines 457  
getNumberOfSearchWordMatches 379  
getNumberOfTableColumns 239  
getNumberOfTableRows 237  
getNumRetries 727  
getNumUsers 65  
getOffset 271  
getOldestDate 441  
getOntModel 682  
getOriginalValue 556  
getOriginalWeight 72  
getOthersRegExp 576  
getOuterXml 637  
getOverallTimeout 727  
getPa 393, 396  
getPageRank 583  
getPageText 625  
getPaginationURLs 262  
getPaginationXPath 264  
getParameters 111, 125  
getParent 494  
getParentNode 238  
getParentURL 405

getPath 65, 148  
getPatterns 265  
getPerformance 126  
getPerformanceCopy 126  
getPerformancesDatasetTrainingFolds 144  
getPerformancesFolds 144  
getPerformancesTrainingFolds 144  
getPhraseFromBeginningOfSentence 492  
getPhraseToEndOfSentence 492  
getPMI 654  
getPostDistribution 627  
getPostGapStandardDeviation 628  
getPower 470  
getPowerDistributionFactor 335  
getPrecision 132, 286  
getPrecisionAt 121  
getPrecisionFor 285  
getPrecisionForCategory 139  
getPredefinedSources 531  
getPreprocessor 123  
getPreviousHeadlines 354  
getPreviousSiblings 504  
getPrint 424, 425  
getPrior 6  
getPriorWeight 57  
getProxy 723  
getProxyList 724  
getPsClassificationStatementConcept 15  
getPsClassificationStatementEntity 15  
getPsFeatureStatement 15  
getPublished 624  
getPublishedSQLTimestamp 624  
getQuery 314  
getQuerySet 241  
getQueryType 241  
getQuestion 568  
getQuestionHashes 402  
getQuestionXPath 399  
getRange 533  
getRangeConcept 537  
getRangeMaxValue 536  
getRangeMinValue 536  
getRangePossValues 536  
getRangeString 532, 537  
getRangeType 537  
getRank 751  
getRankCount 586  
getRanking 587, 746, 747  
getRatedDates 434  
getRawId 624  
getReadableFeedTextType 612  
getReadableBytes 639  
getReadTimeout 727  
getRealCategories 98  
getRealCategoriesString 98  
getRealNumberOfCategory 102  
getRecall 132, 286  
getRecallFor 286  
getRecallForCategory 139  
getReferenceDates 189  
getRegExp 531, 576  
getRegressionRank 580  
getRelativeCorrelation 38  
getRelativeTrust 557  
getRelevance 11  
getReportFolderPath 692  
getRequestCount 739  
getResponseCode 726  
getResultCount 314, 734, 738  
getResultDocument 247  
getResultFilePath 618  
getResultsPerEntity 700  
getResultText 247  
getResultTitle 247  
getRFCRegExp 576  
getRMSE 17  
getRmse 467  
getRootPath 494  
getRootWord 166  
getRuntime 444, 692, 695  
getSafeName 539, 549  
getSafeNewName 532, 539  
getSameDates 434  
getSameDatesMap 434, 435  
getSamePowerFactValues 552  
getSaveType 531  
getSearchEngine 621  
getSearchEngines 712, 739  
getSeedEntities 701  
getSeeds 654

getSensitivityForCategory 140  
getSentence 491  
getSentences 492  
getSeparationString 148  
getSeparator 188, 193  
getSessionDownloadSize 722  
getSettings 43  
getShiftSimilarity 467  
getSiblingPage 719  
getSimilarDocuments 508, 513, 517, 519, 521, 523, 526  
getSimilarity 594  
getSimilarity1 160  
getSimilarity2 160  
getSimilarity3 161  
getSimilarity4 161  
getSimilarity5 161  
getSimilarity6 161  
getSimilarity7 162  
getSimilarity8 162  
getSimilarityReport 508  
getSimilarityThreshold 509  
getSiteUrl 601  
getSketchForDocument 512, 516, 518, 521, 523, 526  
getSketchSize 509  
getSnippetID 658  
getSnippets 545  
getSource 684, 712, 734, 738, 744, 751  
getSourceRetrievalCount 223  
getSourceRetrievalSite 223  
getSources 550, 556  
getSourcesForExtractionType 656  
getSourceURL 658  
getSpecificityForCategory 140  
getSquaredShiftSimilarity 467  
getStandardDeviation 471  
getStatistics 621  
getStemmedTagVocabulary 52  
getStemmer 58  
getStopWords 154  
getStopwords 58  
getString 113  
getStrongInteractionIndicators 347  
getStructureDate 186  
getSubstringBetween 484  
getSummary 751  
getSuperClass 538  
getSurroundingText 354  
getSwitchProxyRequests 724  
getSynonyms 497, 530, 540  
getSynonymsToString 530, 540  
getTableCellPath 235  
getTableName 448  
getTableRows 237, 238  
getTag 200, 209  
getTagAveragedF1 286  
getTagAveragedPrecision 286  
getTagAveragedRecall 286  
getTagConfidenceThreshold 136  
getTagCount 57, 163  
getTagDistance 162  
getTaggedEntities 707  
getTaggedEntryCount 49  
getTaggingFormat 279  
getTaggingType 56  
getTagMatchPattern 57  
getTags 66, 271, 616  
getTagVocabulary 52  
getTargetNode 236  
getTerm1 37  
getTerm2 37  
getTermWeight 9  
getTestDocuments 123, 139  
getTestLimit 51  
getTestSetWeight 7  
getTestStop 49  
getText 35, 274, 311, 580, 625  
getTextByXPath 239  
getTextDump 240  
getTextFeatureType 152  
getTextsByXPath 239  
getTextsByXpath 239  
getTextType 602  
getTfidfThreshold 56  
getThreadCount 229, 723  
getTimeDifferenceToNewestPost 627  
getTimeNewestPost 628  
getTimeOfDay 443  
getTimeOldestPost 627

getTimeRange 627  
getTimestamp 444  
getTimeString 444  
getTimezones 578  
getTitle 233, 309, 365, 601, 623, 751  
getTLD 583  
getTop 116  
getTotalAttributesNumber 659  
getTotalConceptsNumber 659  
getTotalCorrectEntities 696  
getTotalCorrectEntitiesForView 690, 696  
getTotalCorrectFacts 696  
getTotalCorrectFactsForView 691, 696  
getTotalDownloadSize 722  
getTotalEntities 695  
getTotalEntitiesForView 690, 695  
getTotalEntitiesNumber 659  
getTotalEntityPrecision 696  
getTotalEntityPrecisionForView 696  
getTotalFactPrecision 697  
getTotalFactPrecisionForView 697  
getTotalFacts 696  
getTotalFactsForView 691, 696  
getTotalFactsNumber 660  
getTotalSourcesNumber 660  
getTotalTermWeight 7  
getTotalURLStackSize 403  
getTraceResult 523  
getTrainCount 53  
getTrainingDataPercentage 104  
getTrainingDocuments 123, 139  
getTrainingEntities 306, 703, 708  
getTrainingObjects 16  
getTrainingPercentageMax 150  
getTrainingPercentageMin 150  
getTrainingPercentageStep 150  
getTrainingSet 15  
getTrainLimit 50  
getTrainStop 49  
getTrust 227, 228, 350, 549, 558, 582, 706  
getTrustFormula 225  
getType 202, 207, 209, 211, 215, 216, 217, 398, 405, 550  
getTypString 194  
getUnitType 649  
getUnitTypeName 649  
getUnreachableCount 604  
getUnstemMap 53  
getUpdateClass 605  
getURL 184, 589  
getUrl 65, 98, 178, 207, 310, 362, 404, 582, 752  
getURLDate 186  
getURLFromStack 400  
getURLRegExp 577  
getURLs 735, 736  
getURLStackSize 400  
getUserAgent 719  
getValue 495, 553, 555  
getValueCount 532  
getValues 552  
getValueType 530  
getValueTypeByName 529  
getValueTypeName 530  
getValueTypeXSD 531  
getVocByConceptName 348  
getWcm 25, 53  
getWeakInteractionIndicators 348  
getWeakMIOVocabulary 348  
getWebDocument 366, 719, 720  
getWebResults 736  
getWebresults 310, 712  
getWebResultsFromGoogle 736  
getWebServiceIDs 754  
getWeight 71, 495  
getWeightedTerms 99  
getWeightForCategory 140  
getWhat 311  
getWhen 312  
getWhere 311  
getWhitespaces 188  
getWho 311  
getWhy 311  
getWidth 589  
getWidthHeightRatio 590  
getWordDistance 162  
getWorstIndices 662  
getXMLDocument 720  
getXPath 268  
getXPathMap 242  
getXPathSet 263

getYoungestDate 441

GIGA\_BYTES 715

GOOGLE 737

GOOGLE\_8 222

GOOGLE\_BLOGS 738

GOOGLE\_DOMAIN\_PAGE\_RANK 749

GOOGLE\_PAGE\_RANK 749

GradualEntityTrustVoting 84

GREEN 404

greenPrefixCreated 401

GT 559

guessValueType 530

## H

HAKIA 737

HAKIA\_8 221

handleSpecialFormat 649

hasAttribute 541, 542

hasCategoryEntries 705

hasEntity 543

hasEntryWithCategory 10

hashCode 35, 37, 72

hasKeyword 188

hasMaxValue 536

hasMinValue 536

hasNewSynonyms 534, 540

hasPossValue 536

hasSynonym 530, 539

hasVoted 402

hasXMLNS 501

HEAD\_KEYWORDS 562

HeadDate 209

headphone 345

Helper 32

HIERARCHICAL 134

hierarchyRootNode 19

HOUR 205

HOUR\_MS 442

HTML5CanvasExtractor 343

HTMLSymbols 560

htmlToString 463

HTTP\_KEYWORDS 562

HTTPDate 211

## I

ID 172

identifyMIOPages 367

IFM 732

IFrameAnalyzer 344

IllinoisLbjNER 296

IMAGE 227

Image 589

ImageHandler 591

importEntityAssessmentData 32

InCoFiConfiguration 347

increaseAbsoluteCorrelation 38

increaseChecks 603

increaseFrequency 5, 114

increaseNumberOfDocuments 22

increaseThreadCount 230, 723

increaseTotalTermWeight 7

increaseWeight 72

increasNumberOfDocuments 117

increment 426, 427

incrementCount 372

index 23

INDEX\_FILE\_BASE\_PATH 514

init 107, 710

initializeFeatures 350

initialTrust 227

initiateSearch 378

insertRolePage 375

insertRolePageUsage 376

instance 347

isAbsoluteCorrect 554

isActive 713

isAllZero 435

isAlmostCorrect 554

isAnswerHintBeforeAnswer 162

isAudioFile 450

isAutoSave 256

isBenchmark 123, 230, 326, 357, 412

isBigger 648

isBracket 478

isCombineQueries 621

isCompletelyUppercase 479

isContinueQAExtraction 224

isCorrect 546, 553



isCorrectClassified 121  
isDateInRange 179  
isDedicatedPage 352  
isDirtyIndex 54  
isDiscrete 16  
isDownloadPages 642  
isDuplicate 595  
isExtracted 532  
isFAQ 124  
isFastMode 667, 672  
isFeedAutodiscovery 727  
isFileName 450  
isFindNewAttributesAndValues 224  
isForum 123, 124  
isHeadlineTag 464  
isIFrameSource 363  
isInitial 546  
isInMemoryMode 668  
isInstanciated 419  
isLinkedPage 364  
isMainCategory 6  
isNominalClass 16  
isNumber 479  
isNumericExpression 479  
isOnlyPreferred 621  
isRandom 150  
isReadFromIndexForUpdate 24  
isRelevancesUpToDate 8  
isSerialize 126  
isSerializeClassifier 135  
isShowLogging 420  
isSimpleElement 464  
isStopped 230  
isTagBoost 137  
isTagged 283  
isTimeExpression 479  
isURLallowed 359  
isUseAttributeSynonyms 224  
isUseConceptSynonyms 224  
isUseCooccurrence 137  
isUseIndex 21  
isValidURL 725  
isVideoFile 450  
isVowel 480  
isWithinCorrectnessMargin 470

isWithinMargin 470  
isWriteDump 248  
isWriteResultFileContinuously 618

## J

jaccardDistance 509  
jenaDBTest 682

## K

keepXPathPointingTo 234  
KEY\_LOC\_ATTR 201  
KEY\_LOC\_CONTENT 201  
KeywordDate 212  
KEYWORDLOCATION 201  
KeyWords 563  
KILO\_BYTES 714  
KNNClassifier 111  
KnowledgeManager 564

## L

LANGUAGE\_ENGLISH 734  
LANGUAGE\_GERMAN 734  
LanguageDatasetCompiler 759  
learnAndTestClassifierOnline 103  
learnBestClassifier 104  
LEFT 166  
letterNumberCount 480  
limitCategories 99  
limitLinkAnalyzing 346  
LineAction 466  
LINEAR\_REGRESSION 13  
LingPipeNER 298  
LinkAnalyzer 349  
LinkedDataStatisticsCrawler 760  
LinkSetCreator 195  
ListDiscoverer 262  
ListSimilarity 467  
LiveFactExtractor 333  
load 44, 104, 306, 591  
loadAllDictionaries 108  
loadAllRolePagesForConcept 375  
loadConcepts 653

loadConfig 715  
loadDictionary 108  
loadEntities 543, 653  
loadEntity 654  
loadEvaluationEntities 653  
loadNotUsedRolePagesForEntity 375  
loadObjectDescription 389  
loadOntology 653, 681  
loadOntologyFile 681  
loadTrainedClassifier 94  
loadUsedRolePageIdsForEntity 375  
log 103  
LOGGER 606  
LoggerMessage 469  
logMetrics 77  
lowerCaseFirstLetter 476  
lowerCaseText 35  
LT 559  
LUCENE\_INDEX 19

## M

main 17, 27, 32, 34, 36, 44, 47, 58, 63, 76, 79,  
82, 83, 84, 86, 88, 89, 91, 92, 95, 104, 105, 128,  
158, 165, 170, 173, 191, 195, 196, 240, 248, 257,  
264, 276, 295, 297, 299, 301, 303, 305, 307, 314,  
316, 319, 320, 321, 325, 329, 334, 335, 342, 359,  
380, 384, 390, 397, 410, 414, 415, 417, 419, 445,  
449, 460, 464, 474, 485, 495, 498, 499, 506, 510,  
517, 519, 546, 567, 596, 599, 608, 610, 612, 617,  
621, 622, 639, 642, 646, 647, 649, 662, 679, 682,  
688, 693, 701, 707, 728, 730, 733, 736, 739, 742,  
745, 747, 754, 758, 759, 760, 761  
MAJESTIC\_SEO 749  
makeCamelCase 476  
makeContinuousText 481  
makeFullURL 718  
makeMutualXPath 235  
makeRelativeScores 40  
makeSafeName 475  
makeViewName 477  
MapQuery 165  
matches 270  
MathHelper 470  
matrixTest 89

MEASURE\_DAY 438  
MEASURE\_HOUR 438  
MEASURE\_MILLI\_SEC 438  
MEASURE\_MIN 438  
MEASURE\_SEC 438  
MEGA\_BYTES 715  
mergeConcepts 564  
MICROSOFT 737  
MICROSOFT\_8 221  
minEntityCorroboration 232  
minFactCorroboration 232  
MINKOWSKI 591  
MINUTE 205  
MINUTE\_MS 442  
MIO 350  
MIOClassifier 94  
MIOComparator 355  
MIOContextAnalyzer 356  
mioExtractionIsRunning 220  
MIOExtractionProcess 357  
MIOInteractivityAnalyzer 361  
MIOPage 362  
MIOPageCandidateAnalyzer 367  
MIOPageRetriever 368  
mioTypes 346  
MIXX\_VOTES 748  
mobilephone 345  
MONTH 205  
MONTH\_MS 442  
monthNameToNumber 444  
move 457  
movie 345  
MSE 591  
MUC 284  
multAbsRel 12

## N

NAME 172  
NamedEntityRecognizer 277  
NBSP 559  
NBSP2 559  
NEURAL\_NETWORK 13  
NewsAggregator 640  
NewsAggregatorException 643

NGram 113  
NGramIndex 116  
NL 560  
NO\_CORRELATIONS 60  
nodeInBox 236  
nodeInTable 235  
NoisyOr 86  
NON\_RED 404  
normalize 43  
normalizeAllEntities 257  
normalizeDate 645  
normalizeDateFormat 645  
normalizeName 546  
normalizeNumber 647  
normalizeTag 757  
normalizeYear 192  
numberCount 480  
NumericFactDistribution 335

## O

oneFullDayHasBeenSeen 605  
OOUML 560  
OpenCalaisNER 301  
openIndex 511, 514, 519, 520, 523, 525  
OpenNLPNER 302  
openReader 667, 672, 674, 676  
openWriter 668, 672, 674, 676  
orderDates 440, 441  
orderDatesArray 441  
orderHashMap 435  
OUML 560  
overlap 471  
overlaps 270

## P

PageAnalyzer 233  
PageContentExtractor 245  
PageContentExtractorException 249  
PATTERN\_PHRASE 226  
performAction 466, 630  
performActionOnEveryLine 453  
performLinearRegression 472  
perm 506

PersistenceManager 683  
PhraseChunker 320  
PhraseExtractor 265  
PLAIN 411  
PMI 87  
POSSIBLE 285  
PredefinedSource 684  
PreflightFilter 250  
PrefuseGraph 710  
prependFile 454  
preProcessDocument 119  
preprocessDocument 105, 109, 110, 112, 124, 125, 128  
Preprocessor 118  
preProcessPage 119, 120  
preProcessString 119  
preProcessText 120  
PRESET\_INTENSE\_EVALUATION 149  
PRESET\_MODERATE\_EVALUATION 149  
PRESET\_SIMPLE\_EVALUATION 149  
print 424, 425  
printDateArray 432, 433  
printDateMap 433  
printDOM 240  
printer 345  
printEvaluationDetails 279  
printEvaluationFiles 147  
printExtractions 256  
printStatistics 49  
put 116, 447  
putArticleInFront 482  
putDataObject 428  
putFeature 626

## Q

QA 568  
QA\_SITE 398  
qaExtractionIsRunning 220  
QAExtractionProcess 392  
QAExtractionThread 393  
QASite 398  
QASites 403  
QAUrl 404  
QAUrlStack 406

- QUANTITY\_TRUST 220
- Query 241
- QueryWord 166
- QuicktimeExtractor 370
- QUOT 559
- QuoteCrawler 761
- 
- R
- 
- RandomGraphWalk 89
- RANGETYPE\_MINMAX 536
- RANGETYPE\_POSS 536
- RANK\_AVERAGE 731
- RankAggregation 731
- rankAnswer 158
- RATE 214
- read 62, 69, 487, 665, 669, 673, 675
- read1 665, 669
- read3 665, 670
- readFeatureObjects 14
- readFileToArray 452
- readFileToString 452
- readHTMLFileToString 451
- readTest 69
- readTrain 69
- RecognizedEntities 704
- RecognizedEntity 705
- REDDIT\_SCORE 748
- redoWeak 346
- ReferenceDate 214
- RegExp 575
- remove 447
- removeAll 638
- removeAnchors 719
- removeAttribute 542, 682
- removeBrackets 478
- removeConcept 565, 681
- removeConcreteHTMLTag 462
- removeControlCharacters 481
- removeCrawlerCallback 723
- removeDoubleWhitespaces 486
- removeDuplicateLines 453
- removeDuplicates 411
- removeFirstStringpart 485
- removeFormat 433
- removeHTMLTags 462
- removeLastWhitespace 486
- removeNodigits 192
- removeNonAsciiCharacters 478
- removeNullElements 422
- removeNumbering 477
- removeRange 533
- removeRangeValue 533, 537
- removeSiblingPagePaths 263
- removeSource 556
- removeSpecialChars 478
- removeStopWords 477
- removeTimezone 194
- removeUnrelevantRolePages 376
- removeURLFromStack 401
- removeWhitespace 636
- removeXPathIndices 239, 240
- removeXPathIndicesNot 240
- rename 455
- replaceHTMLSymbols 464
- Report 690
- ReportFileParser 694
- ReportSet 695
- rescaleImage 592
- rescaleImage2 593
- rescaleImage3 593
- rescaleImage\_broken 593
- rescaleImageOptimal 592
- reset 122
- resetWeights 494
- resultCount 345
- retrieveHitCounts 82
- RETRIEVAL\_EXTRACTION\_TYPE\_FOCUSED\_CRAWL 258
- RETRIEVAL\_EXTRACTION\_TYPE\_PHRASE 258
- RETRIEVAL\_EXTRACTION\_TYPE\_SEED 258
- retrieveMIOPages 368
- reverse 424
- reverseString 483
- RIGHT 166
- RolePage 372
- rolePageRelevanceValue 346
- rolePageTrustLimit 346
- round 470
- run 253, 254, 323, 326, 340, 357, 392, 393, 408,

412, 713  
runFactExtractionBenchmark 222  
runQAFromOfflineTestset 396  
runQuery 660, 661  
runUpdate 661, 662  
runVoting 75, 76, 81, 83, 84

## S

sameTag 271  
save 44, 108, 111, 127, 273  
saveAsCSV 22  
saveCompleteReportSet 698  
saveDictionary 108  
saveExtractions 567, 654, 681, 683  
saveImage 595  
saveImage2 595  
saveImage3 595  
saveIndex 510, 511, 515, 519, 523  
saveToFile 619  
saveTotalOnly 698  
saveTrainedClassifier 95  
saveURLDump 716  
scoreNER 299  
SearchAgent 378  
searchEngine 346  
searchFeeds 730  
SearchWordMatcher 379  
SECOND 205  
SECOND\_MS 442  
SECOND\_PRIORITY 562  
secondPriorityKeywords 563  
SELECTION 172  
SELECTION\_HALF 172  
separateFile 155  
serialize 23, 403, 455, 564  
sessionDownloadedBytes 715  
set 200, 203, 207, 214  
setAbsoluteCorrelation 38  
setActive 713  
setAdded 602, 624  
setAll 207  
setAllAnswersXPath 400  
setAllFalse 185  
setAllowedFiletypes 67  
setAllTrue 185  
setAltText 354  
setAnswerClassifier 395  
setAnswerHintBeforeAnswer 162  
setAnswerPrefix 400  
setAnswerSuffix 400  
setAnswerWordCount 160  
setAssignments 287  
setAttribute 325, 552, 677  
setAttributeNames 685  
setAttributes 541  
setAutoSave 256  
setAverageTagOccurence 53  
setBenchmark 123, 230, 326, 357, 412  
setBenchmarkSet 225  
setBenchmarkSetSize 224  
setBenchmarkType 225  
setBestAnswerXPath 399  
setCacheTtlSeconds 747  
setCategories 23, 122, 138, 674  
setCategory 11  
setCategoryEntries 12, 705  
setCheckApproach 607  
setCheckInterval 607  
setChecks 603  
setChildren 494  
setChosenClassifier 16  
setClassAssociation 30  
setClassificationType 134, 138  
setClassificationTypeSetting 125, 143  
setClassificationTypeTagSetting 135  
setClassifiedAs 100  
setClassifier 17, 143  
setClassType 6, 24  
setCombineQueries 621  
setConcept 545, 677  
setConceptID 373  
setConceptName 684  
setConcepts 257  
setConnectionTimeout 726  
setContinueQAExtraction 224  
setCorrectValue 553  
setCorrectValues 566  
setCorrelationType 56  
setCorrelationWeight 57

setCount 373  
setCountDown 473  
setCrawler 393  
setCurrentSource 323  
setDatabaseType 24  
setDataPath 63  
setDataSetLocation 701  
setDatasetName 700  
setDatasets 151  
setDateString 206  
setDbDriver 447, 666, 671  
setDbHost 448, 666, 671  
setDbName 667, 671  
setDbPassword 448, 667, 672  
setDbPort 448, 666, 671  
setDbType 447, 666, 671  
setDbUsername 448, 667, 672  
setDebugDump 619  
setDedicatedPage 352  
setDedicatedPageTrust 365  
setDescription 678  
setDictionary 109, 676  
setDirectURL 351  
setDirtyIndex 54  
setDiscrete 16  
setDistinctTagCount 163  
setDocument 233, 245, 246, 324, 719  
setDocumentType 100  
setDownloadPages 642  
setDuplicateCount 586  
setEntities 543  
setEntity 271, 324, 351, 416  
setEntityChunks 310  
setEntityFeatures 310  
setEntityName 333  
setEntries 603  
setEntryText 625  
setEntryURL 399  
setEvaluation 17  
setEvaluationSetting 146  
setEvents 313  
setExtractedAt 550, 557  
setExtractionLimit 257  
setExtractionType 330, 582  
setFact 555  
setFacts 545  
setFactString 330  
setFactValue 582  
setFastMode 667, 672  
setFeature 353, 579  
setFeatureNames 30  
setFeatures 30, 310, 318, 354, 356, 415, 625  
setFeatureSetting 126, 143  
setFeedAutodiscovery 727  
setFeedId 623  
setFeedProcessingAction 607  
setFeedUrl 601  
setFileName 353  
setFileSize 353  
setFilter 63  
setFindNewAttributesAndValues 224  
setFindPageURL 351  
setFormat 207, 602  
setFrequency 114  
setFvWekaAttributes 15  
setGreenPrefix 401  
setGreenPrefixCreated 401  
setGreenUrlDepth 402  
setHeight 590  
setHostname 373  
setHow 312  
setId 373, 538, 549, 582  
setId 601, 623  
setIdf 114  
setIdfCount 53  
setIdfIndex 52  
setIframeParentPage 365  
setIframeParentPageTitle 365  
setIframeSource 363  
setIgnores 620  
setImageContent 590  
setIndex 114  
setIndexedPrior 6  
setIndexName 511, 514, 523  
setIndexPath 24, 676  
setIndexType 24  
setInitial 546  
setInMemoryMode 668  
setInteractivityGrade 352, 361  
setK 111

setKbCommunicator 307  
setKeyword 212  
setkFolds 150  
setKnowledgeManager 229, 540  
setLabel 494  
setLanguage 602, 736  
setLastChecked 605  
setLastDownloadSize 722  
setLastFeedEntry 604  
setLastHeadlines 604  
setLastSearched 541, 549  
setLength 271  
setLink 624  
setLinkedPage 364  
setLinkName 363  
setLinkParentPage 364  
setLinkTitle 364  
setLoggerName 469  
setLongestPostGap 628  
setMainCategories 21  
setMainCategory 6  
setMainContent 583  
setMaxCheckInterval 603  
setMaxFails 742  
setMaxFileSize 67  
setMaximumTermLength 153  
setMaximumURLs 399  
setMaxNGramLength 153  
setMaxTags 137  
setMaxTerms 153  
setMaxThreads 313, 620, 641, 723, 742  
setMedianPostGap 628  
setMessage 469  
setMeticulousPostDistribution 604  
setMinCheckInterval 603  
setMinimumTermLength 153  
setMinNGramLength 153  
setMinTags 136  
setMinUsers 67  
setMinUserTagRatio 67  
setMIOType 352  
setN 113  
setName 5, 23, 71, 125, 279, 398, 539, 549, 705  
setNearestTextkeyword 189  
setNewName 532, 539  
setNewSuperClass 539  
setNewSynonyms 534, 540  
setnGramLength 509  
setNominalClass 16  
setNumberOfDocuments 22, 117  
setNumRetries 727  
setOffset 271  
setOnlyPreferred 620  
setOriginalValue 556  
setOverallTimeout 727  
setPa 393, 397  
setPageText 625  
setPaginationXPath 264  
setParent 494  
setParentURL 405  
setPath 148  
setPerformance 126  
setPerformancesDatasetTrainingFolds 144  
setPerformancesFolds 144  
setPerformancesTrainingFolds 144  
setPostDistribution 627  
setPostGapStandardDeviation 628  
setPrecision 132  
setPredefinedSources 531  
setPreprocessor 123  
setPreviousHeadlines 354  
setPriorWeight 57  
setProxy 723  
setProxyList 724  
setPsClassificationStatementConcept 15  
setPsClassificationStatementEntity 15  
setPsFeatureStatement 15  
setPublished 624  
setQuery 314  
setQuerySet 241  
setQueryType 241  
setQuestion 568  
setQuestionHashes 402  
setQuestionXPath 399  
setRandom 150  
setRangeType 537  
setRankCount 586  
setRat 180  
setRateToZero 180  
setRateWhightedByGroups 180

setRawId 624  
setReadFromIndexForUpdate 24  
setReadTimeout 727  
setRealCategories 97  
setRecall 132  
setReferneceLookUp 178  
setRegExp 531  
setRelativeCorrelation 38  
setRelevancesInPercent 9  
setRelevancesUpToDate 8  
setResultCount 314, 734, 738  
setResultFilePath 618  
setResultLimit 620  
setResultsPerEntity 700  
setRmse 467  
setRootWord 166  
setRuntime 692, 695  
setSaveType 531  
setSearchEngine 621  
setSeedEntities 701  
setSeparationString 148  
setSerializeClassifier 135  
setServices 746  
setSettings 44  
setShiftSimilartiy 467  
setShowLogging 420  
setSimilarity1 160  
setSimilarity2 161  
setSimilarity3 161  
setSimilarity4 161  
setSimilarity5 161  
setSimilarity6 161  
setSimilarity7 162  
setSimilarity8 162  
setSimilarityThreshold 509  
setSiteUrl 601  
setSketchSize 509  
setSource 684, 735, 738  
setSources 550, 556  
setSquaredShiftSimilartiy 467  
setStemmedTagVocabulary 52  
setStemmer 58  
setStopCount 716  
setStopped 230  
setStopWords 154  
setStopwords 58  
setString 113  
setSuperClass 538  
setSurroundingText 354  
setSwitchProxyRequests 724  
setSynonyms 530, 540  
setTableName 448  
setTag 199, 210  
setTagBoost 137  
setTagConfidenceThreshold 136  
setTagCount 57, 163  
setTagDistance 162  
setTagged 283  
setTaggingFormat 279  
setTaggingType 56  
setTagMatchPattern 57  
setTags 271  
setTagVocabulary 52  
setTechArchive 185  
setTechHTMLContent 184  
setTechHTMLHead 184  
setTechHTMLStruct 184  
setTechHTTP 184  
setTechReference 184  
setTechURL 184  
setTestDocuments 123, 139  
setTestField 655  
setTestLimit 51, 70  
setTestSetWeight 7  
setText 311  
setTextFeatureType 152  
setTextType 603  
setTfidfThreshold 56  
setTimeNewestPost 628  
setTimeOldestPost 628  
setTitle 310, 365, 602, 624  
setTotalDownloadSize 722  
setTrainCount 53  
setTrainingDataPercentage 104  
setTrainingDocuments 123, 138  
setTrainingObjects 16  
setTrainingPercentageMax 150  
setTrainingPercentageMin 150  
setTrainingPercentageStep 150  
setTrainingSet 15



setTrainLimit 51, 70  
setTrust 350, 549, 558, 582, 706  
setTrustFormula 225  
setTtlSeconds 745  
setType 399, 405, 550  
setUnreachableCount 604  
setUnstemMap 53  
setUpdateClass 605  
setURL 184, 589  
setUrl 98, 178, 207, 310, 362, 405, 582  
setUseAttributeSynonyms 224  
setUseCompression 726  
setUseConceptSynonyms 224  
setUseCooccurrence 137  
setValue 495, 555  
setValueCount 532  
setValues 552  
setValueType 531  
setVoted 402  
setWcm 25, 53  
setWebresults 311  
setWeight 71, 495  
setWeightedTerms 100  
setWhat 311  
setWhen 312  
setWhere 311  
setWho 311  
setWhy 312  
setWidth 589  
setWordDistance 162  
setWordPair 37  
setWriteDump 248  
setWriteResultFileContinuously 618  
setX 385  
setXPath 268  
setY 385  
sha1 484  
SHALLOW\_CORRELATIONS 60  
SHINGLES 411  
Shingles 507  
ShinglesIndexBaseImpl 514  
ShinglesIndexH2 516  
ShinglesIndexJava 518  
ShinglesIndexJDBM 520  
ShinglesIndexTracer 522  
ShinglesIndexWB 525  
showBits 506  
showGraph 710  
showTestDocuments 127  
showTrainingDocuments 126  
SilverlightExtractor 381  
SINGLE 134  
size 448  
SLASHES 280  
slashToColumn 275  
slashToXML 275  
sleep 489  
Snippet 579  
SnippetBuilder 409  
SnippetClassifier 169  
SnippetDuplicateDetection 411  
snippetExists 658  
snippetExtractionIsRunning 220  
SnippetExtractionProcess 412  
SnippetFeatureExtractor 415  
SnippetQuery 416  
sort 273  
sortByRelevance 9  
sortByValue 423  
sortCategoriesByRelevance 98  
Source 581  
SOURCE\_TRUST 220  
SourceAggregator 732  
SourceRetriever 734  
Sources 584  
SPECIAL\_MARKER 284  
StanfordNER 304  
start 473, 741  
startContinuousReading 606  
startCrawl 389, 716  
startElement 250  
startEntityExtraction 222  
startExtraction 255, 316, 328, 359, 395, 414  
startFactExtraction 222  
startFullExtractionLoop 223  
startMIOExtraction 223  
startQAExtraction 223  
startSnippetExtraction 223  
startsUppercase 480  
startsWithEntity 580

startTest 47, 70  
startTesting 49  
startTrain 47, 70  
startTraining 49  
stop 64, 473  
STOP\_DAY 437  
STOP\_HOUR 437  
STOP\_MINUTE 437  
STOP\_MONTH 437  
STOP\_SECOND 437  
STOP\_WORDS\_DE 33  
STOP\_WORDS\_EN 33  
STOP\_YEAR 437  
stopContinuousReading 606  
stopEntityExtraction 222  
stopExtraction 230, 253, 326, 357, 392, 412  
stopFactExtraction 222  
stopMIOExtraction 223  
stopQAExtraction 223  
stopSnippetExtraction 223  
stopTesting 49  
stopTraining 49  
StopWatch 473  
Stopwords 33  
StringHelper 475  
StringInputStream 487  
StringNormalizer 647  
StringOutputStream 488  
StringTagger 707  
stringToXml 637  
strongInteractionIndicators 347  
STRUCTURE\_DEPTH 199  
STRUCTURED\_PHRASE 226  
StructureDate 216  
SVM 13  
SVM2 13  
SWFContentAnalyzer 382  
SZLIG 560

**T**

TABLE\_CELL 226  
TAG 134  
Tag 71  
tag 43, 278, 295, 301  
tagAndSaveString 707  
TagComparator 73  
tagDefineFont2 382  
tagDefineFontInfo 382  
tagDefineText 384  
tagDefineTextField 383  
tagString 707  
TECH\_ARCHIVE 176, 205  
TECH\_HTML\_CONT 175, 204  
TECH\_HTML\_HEAD 175, 204  
TECH\_HTML\_STRUC 175, 204  
TECH\_HTTP\_HEADER 175, 204  
TECH\_REFERENCE 175, 205  
TECH\_URL 175, 204  
tempDirPath 346  
Term 35  
TEST 97  
test 46, 70  
testClassifier 14, 103, 158  
testCrawler 196  
TestDocument 121  
TESTING 548  
TestKnowledgeBaseCommunicator 708  
testNER 297  
testProcedure 662  
text 385  
TEXT\_TYPE\_FULL 600  
TEXT\_TYPE\_NONE 600  
TEXT\_TYPE\_PARTIAL 600  
TEXT\_TYPE\_UNDETERMINED 600  
TextClassifier 122  
TextDumper 385  
TEXTRUNNER 738  
THIRD\_PRIORITY 562  
thirdPriorityKexwords 563  
ThreadHelper 489  
THRESHOLD 59  
timelsUp 473  
toDouble 475  
toGrayScale 594  
toHashSet 425  
toInt 475  
tokenize 490  
Tokenizer 490  
toList 691

toString 7, 12, 24, 31, 34, 35, 38, 40, 44, 49, 51,  
54, 58, 65, 72, 100, 114, 127, 135, 137, 148, 151,  
154, 200, 202, 207, 212, 217, 272, 287, 330, 402,  
426, 488, 495, 532, 543, 546, 554, 558, 568, 580,  
583, 587, 605, 626, 628, 706, 752  
totalCorrectEntities 689  
totalCorrectFacts 689  
totalEntities 689  
totalFacts 689  
train 42, 46, 69, 277, 278, 294, 296, 298, 301,  
303, 304, 306  
trainAndTestClassifier 103  
trainClassifier 14, 78, 94, 103, 158, 169  
TRAINING 97, 548  
TrainingDataSeparation 155  
trainNER 297, 298, 305  
transformRelevancesInPercent 9  
transformToEvaluationAnnotations 273  
TreeNode 493  
trim 481  
tsvToSsv 276  
TUDNER 306  
TWEETS 749  
TWITTER 738  
TYPE\_BROWSE\_XP 259  
TYPE\_INDEX\_OF\_XP 259  
TYPE\_LIST\_OF\_XP 259  
TYPE\_SEED\_2 259  
TYPE\_SEED\_3 259  
TYPE\_SEED\_4 259  
TYPE\_SEED\_5 260  
TYPE\_XP\_ESPECIALLY 259  
TYPE\_XP\_INCLUDING 258  
TYPE\_XP\_LIKE 258  
TYPE\_XP\_SUCH\_AS 258  
TYPE\_XS\_INDEX 259  
TYPE\_XS\_LIST 259  
  
U  
  
UNKNOWN 404  
UNCLASSIFIED 97  
UNIT\_DIGITAL 535  
UNIT\_FREQUENCY 535  
UNIT\_LENGTH 535  
UNIT\_TIME 535  
UNIT\_UNITLESS 535  
UNIT\_WEIGHT 535  
unitLookup 648  
UnitNormalizer 648  
unitsSameType 648  
UniversalMIOExtractor 387  
UNKNOWN 226, 548  
unzipFile 458  
unzipFile7z 459  
unzipFileCmd 459  
unzipFileToString 459  
unzipInputStreamToString 459  
update 419, 665, 670, 673, 675  
updateChartsOnly 692  
updateCheckIntervals 607  
updateExtractionStatus 655  
updateFeed 614, 631, 635, 640  
updateFeedPostDistribution 614  
updateNegativePrefix 401  
updateOntology 652  
updateOntologyFile 681  
updatePair 39  
updatePositivePrefixes 401  
updateRolePage 376  
updateTrust 566  
updateWCM 21  
updateWord 21  
upperCaseFirstLetter 476  
URL\_BINARY\_BLACKLIST 229  
URL\_TEXTUAL\_BLACKLIST 229  
URLClassifier 128  
URLDate 217  
urlDecode 485  
URLDownloader 741  
urlEncode 485  
URLRankingCache 744  
URLRankingServices 746  
URLs 130  
urlsAvailable 401  
URLStack 750  
Urns 91  
useIndex 20, 107  
useLearnedNER 297, 299, 305  
useMemory 20, 107

USER\_INPUT 226  
useTrainedClassifier 158, 169, 321  
UUML 560  
UUUML 560

## V

VALUE\_AUDIO 529  
VALUE\_BOOLEAN 528  
VALUE\_DATE 528  
VALUE\_IMAGE 528  
VALUE\_MIXED 529  
VALUE\_NUMERIC 528  
VALUE\_STRING 528  
VALUE\_URI 529  
VALUE\_VIDEO 529  
valueOf 59, 60, 281, 598, 749  
values 59, 60, 280, 598, 749  
verifyURL 726  
VERSION 172

## W

weakInteractionIndicators 346  
weakMIOs 345  
WEB 172  
WebResult 751  
WEBRESULT\_SUMMARY 409  
WEEK\_MS 442  
WEIGHT\_BODY\_TERM 118  
WEIGHT\_DOMAIN\_TERM 118  
WEIGHT\_KEYWORD\_TERM 118  
WEIGHT\_META\_TERM 118  
WEIGHT\_TITLE\_TERM 118  
WhereClassifier 321  
WORD\_NGRAMS 152  
WordCorrelation 37  
WordCorrelationMatrix 39  
WordFeatureClassifier 92  
WordNet 497  
wordToPlural 499  
wordToSingular 499  
WordTransformer 499  
WrapperInductor 266  
write 487, 488, 665, 670, 673, 675

write3 666, 670  
writeCSV 318  
writeDataToReport 44  
writeIndex 679  
writeToFile 453, 454  
writeXmlDump 636  
WSW 754

## X

XML 280  
xmlToColumn 275  
xmlToString 636  
XPathAffixWrapper 268  
XPathHelper 501  
XPathSet 242  
XY\_LINE\_CHART 687  
XY\_SCATTER\_CHART 687

## Y

YAHOO 737  
YAHOO\_8 221  
YAHOO\_BOSS 737  
YAHOO\_BOSS\_NEWS 738  
YAHOO\_DOMAIN\_LINKS 748  
YAHOO\_PAGE\_LINKS 749  
YEAR 205  
YEAR\_MS 443  
YELLOW 404

## Z

zip 458  
zipString 458