2012

@Note2 – Plug-ins Guide

Hugo Costa and Miguel Rocha

University of Minho, BisBII Group

28-03-2012

This document serves to explain the basic steps of @Note2 development. We present the @Note2 plug-ins created under the MVC AIBench Model. For more information about AIBench please visit http://www.aibench.org/.

For more details or suggestion please contact us:

Miguel Rocha (mrocha@di.uminho.pt)

Hugo Costa (hcosta@di.uminho.pt)

Index

Development

In this section we present some details regarding @Note2 development. The @Note2 is divided into modular software units called plug-ins (set of software components that add specific abilities to the application). These plugins are AIBench-based and have dependencies between them.

Plug-ins

Anote2Core

This plug-in contains the basic interfaces to support the functionalities of @Note2. It is organized in three independent units: *core*, *process* and *resources*.

- The *core* unit contains interfaces for database connection, documents and annotation handling and basic configurations (See Figure 1).
- The *process* unit contains interfaces for biomedical text mining processes: information retrieval and information extraction.
    - The information retrieval component is further divided in search and crawling processes.
    - The information extraction unit is divided in: Named Entity Recognition (NER) and Relation Extraction (RE) processes (See Figure 1).
- The *resource* unit provides interfaces for lexical resources. Currently, these include dictionaries (with appropriate loaders), lookup tables, syntactic rules and ontologies. For resource management there are available resource elements and resource elements set.

Figure 1 - Anote2Core Interfaces Diagram.

Configuration

Figure 2 illustrates the AIBench default configuration file present in each plug-in. Each source folder must have a configuration file.

```
<plugin start="true">
    <uid>pt.uminho.anote.core</uid>
    <name>anote core</name>
    <version>2.0.0</version>
</plugin>
```

Figure 2 – Anote2Core AIBench configuration File

Anote2DataStructures

The Anote2DataStructures plug-in contains data structures that implement ANote2Core interfaces. This is organized into the following packages given below:

Annotations

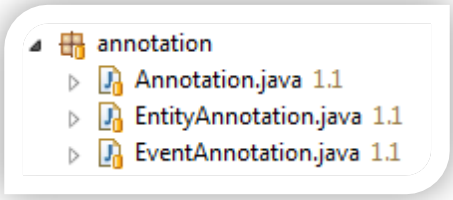

Figure 2 - *Annotations package in* Anote2DataStructures *plug-in*

Package that implements text annotations. These are divided in Entity Annotations, Event Annotations and Text Structuring Annotations.

Annotation: Represents a generic annotation (keeping features such as start and end offsets, annotation type and database identifier).

EntityAnnotation: Represents an entity annotation: Each annotation includes the entity name, classification (Ontological term ID in database) and standardized form.

EventAnnotation: Represents an event annotation. Each annotation is composed of the left entities list, right entities list, event clue and ontology database identifier (for event classification).

Configuration

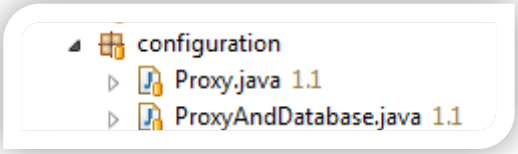

Figure 3 – *configuration package* in Anote2DataStructures plug-in

This is the package that implements the necessary settings for @Note to run properly. The settings are saved in AIBench /conf/settings.conf file. Settings are composed for Proxy and Database:

Proxy: Save proxy settings like proxy status (active or not), port and host.

ProxyAndDatabase: Contains the proxy and database settings for plug-in access.

Database

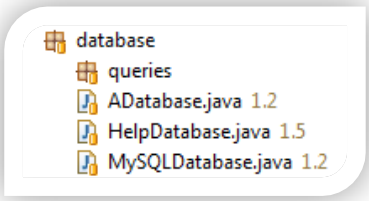

Figure 4 - *database* package in Anote2DataStructures plug-in.

Package that contains database information. Database credentials are saved in AIBench configuration file */conf/settings.conf*.

Database: Abstract class that implements the settings for database access: Host, port, schema, user, password and connection.

HelpDatabase: Class that helps in database linkage.

MySQLDatabase: Implementation of MySQL database and methods to open and get database connection.

Queries: Module that implements SQL queries.

Documents

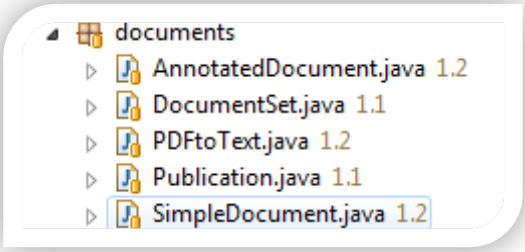

Figure 5 - *documents* package for Anote2DataStructures plug-in

Package for document representation. Documents could be simple or have semantic annotations (event /entities annotations).

AnnotatedDocument: Represents a document that contains entity and/or event annotations. Defines the annotation process and Corpus for the document. Each publication may be in different Corpus and processes.

DocumentSet: Represents a Set of Documents.

PDFtoText: Class that saves the method for converting a PDF File (given file path) into a text stream.

Publication: Represents a specific document, such as a publication. Extends *SimpleDocument* and contains information about PMID (PubMed identifier), journal, journal url and available PDF (and more specific fields).

SimpleDocument: Represents a basic document. Contains information about database identifier, title, and authors, abstract and content (*String*).

Process

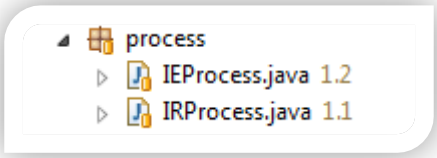

Figure 6 –*process package in* Anote2DataStructures plug-in

Package for handling Biomedical Text Mining processes. Processes are pipelines that allow, in the case of an *IRProcess* for the creation of a document set and in the case of an *IEProcess* for the creation of sets of annotation over sets of documents (Corpus)

IRProcess: Represents the base model of a process of information retrieval.

IEProcess: Represents the simple model of a process of information extraction.

Resources

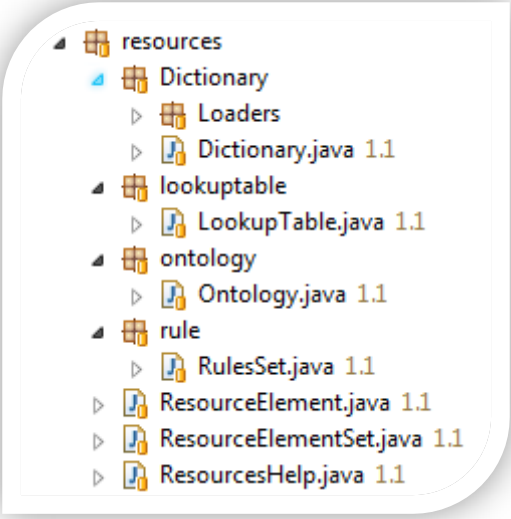

Figure 7 – *resources package in* Anote2DataStructures plug-in

Package for lexical resources. Resources can be defined as lists of elements structured so that they can be used in Biomedical Text Mining processes (e.g. dictionaries to identify entities in text or ontologies for the classification of those entities). In this package there are dictionaries, lookup tables, ontologies and syntactic rules.

Dictionary: Represents a dictionary, a list of terms and synonyms. Keeps information about name, description and database ID.

LookupTable: Represents a term list, often generated manually.

Ontology: Represents an ontology.

RulesSet: Represents a set of syntax rules that can be applied in information extraction processes.

ResourceElement: Represents a resource element. Keeps information about database ID, designation, classification, links to external database, term origin.

ResourceelementSet: Resource element Set

ResourceHelp: Class with methods to support resource management.

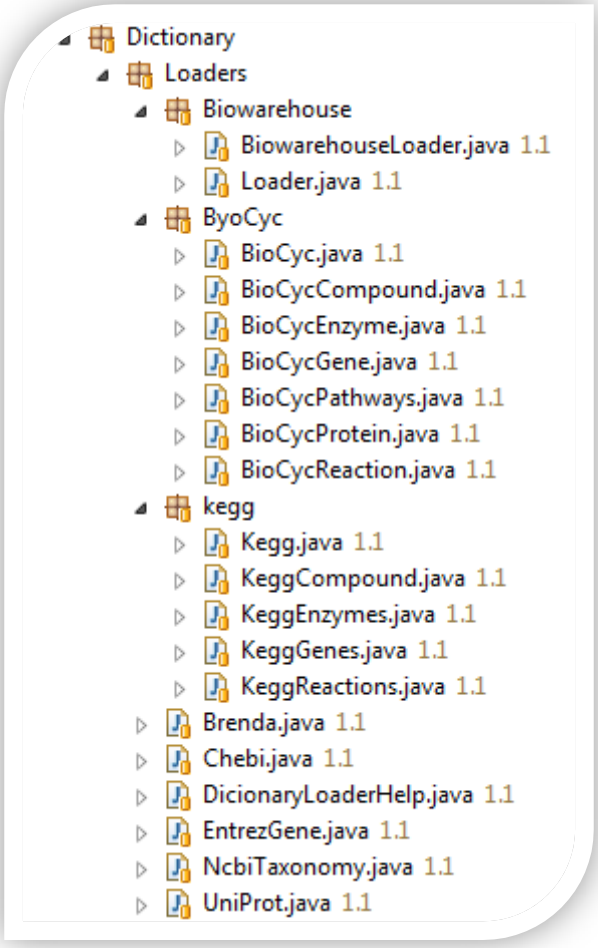

Figure 8 – Dictionary loaders in Anote2DataStructures

Dictionary loaders (Figure 8) are available for database flat files (supports Biocyc, KEGG, Brenda, Chebi, Entrezgene, NCBI taxonomy and Uniprot). There is also available a loader for the BioWarehouse Database.

TextProcessing

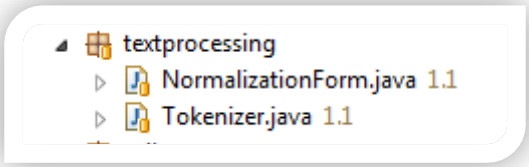

Figure 9 – *textprocessing package in* Anote2DataStructures plug-in

`NormalizationForm`: Class containing methods for text normalization.

`Tokeniser`: Class containing the simple tokenizer developed in-house splitting the text into segments (words or punctuation marks).

Utils

Package that contains some utilities:

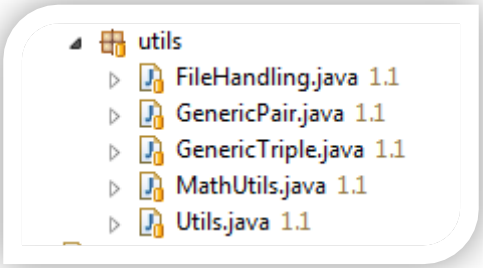

Figure 11 - *utils* package in Anote2DataStructures plug-in

`Filehandling`: Class with methods for manipulating files and directories.

`GenericPair`: Class with the implementation for pair data structure.

`GenericTriple`: Class with the implementation of a triple data structure.

`MathUtils`: Class with the implementation of some mathematical functions.

`Utils`: Class with other useful methods.

Configuration

```
<plugin start="true">
    <uid>pt.uminho.anote.datastructures</uid>
    <name>anote data structures</name>
    <version>2.0.0</version>

    <dependencies>
        <dependency uid="pt.uminho.anote.core"/>
    </dependencies>

</plugin>
```

Figure 10 - *Anote2DataStructures plug-in* AIbench configuration file.

Anote2PubmedRetrievalUI

The Information Retrieval plug-in is divided into two main components:

1. Publication searching in PubMed, supporting search using a combination of keywords and specific fields. The result is a set of publication that contains information about title, abstract, journal.

2. These publications are organized in queries and Publication Retrieval tries to get the PDF file for a Publication indexed by its PMID.

There is a limitation for crawling that is limited to articles with free access only, but this option can be changed depending on the organization's rights. The is also a system for classification of publications within a query given its relevance.

This plugin was developed in collaboration with the SING group at University of Vigo - Spain.

The main classes are the following (organized according to the MVC model of AI Bench):

Data-types

PublicationManager: main data-type of the plug-in that contains information about all PubMed searches already executed. Contains information about proxy and database given by configuration file and directory for PDF files.

QueryInformationRetrievalExtension: represents a Query. Contains information about database ID, date, keywords, organism, matching publications, available abstracts and other generic query properties.

Operations

AddFileToPublicationManagerOperation: Manually adds a PDFfile for a publication.

AddPublicationToQueryOperation: Adds a new publication to a *QueryInformationRetrievalExtension*.

ExitOperation: Publication Manager exit operation.

InitReferenceManager: Initialize *PublicationManager* plug-in.

JournalRetrivalListDocs: Operation that given a list of PMIDs tries to get PDF files.

PubmedSearchOperation: Operation for searching in PubMed, given the query details.

SelectRelevance: Change document relevance for query.

UpdateQueryOperation: Update *QueryInformationRetrievalExtension* (updates the result of PubMed search in time).

Views

PublicationManagerView: `PublicationManager` View that contains a visualizer of all queries present in the Publication Manager and allows applying some filters.

`QueryRelevanceView: QueryInformationRetrievalExtension` View that permits viewing documents in a query including relevance.

`QueryView: QueryInformationRetrievalExtension` View that permits viewing documents in a query and permits some search steps.

Configuration

```
<plugin start="true">
    <uid>pt.uminho.anote.aibench.referencemanager</uid>
    <name>anote publication manager</name>
    <version>2.0.0</version>


    <dependencies>
        <dependency uid="pt.uminho.anote.core"/>
        <dependency uid="pt.uminho.anote.datastructures"/>
        <dependency uid="pt.uminho.anote.aibench.utils"/>
    </dependencies>
```

Figure 12 - Anote2PubmedRetrievalUI AIbench configuration file.

Anote2CorporaUI

Plug-in that defines central data-types for Corpora (Corpus Set). All information extraction processes are applied over a Corpus in @Note2. A Corpus is a set of documents that could be annotated with entities/events in IEProcesses.

The main classes are the following (organized according to the MVC model of AI Bench):

Datatypes

Corpora: Represents Corpora and contains a Corpus Set. Contains methods for Corpus database management.

Corpus: Represents a set of publications. Contains information about Corpus properties, name, description, database id and lists of IEPRocess applied to corpus.

NERDocumentAnnotation: Contains information about document entities annotations resulting from NER processes.

NERProcess: Represents a NER Process and contains a set of *NERDocumentAnnotation*.

REProcess Represents a RE Process and contains a set of REDocumentAnnotations

REDocumentAnnotation: Contains information about document entities and event annotation resulting from RE processes.

Operations

ChangeClassColor: Operation for changing class color. The color serves to view multi-colors.

CreateCorpusOperationByPublicationManager: Operation that permits corpus creation deriving for Queries of Publication Manager.

ExitOperation: Plug-in exit operation

InitProject: Plug-in start operation

Views

CorporaView: Allows the visualization of Corpus data-types on the clipboard.

CorpusDocumentsView: Allows the visualization of the documents belonging to each corpus.

CorpusProcessesView: Allows the visualization of the processes applied to each corpus.

NERAnnotatedDocumentView: NERDocumentAnnotation View; allows checking the document entity annotations.

NERProcessAnnotationDocumentsView: *NERProcess* View of all document and the creation of *NERDocumentAnnotation*.

NEREntityStatisticsView: *NERProcess* View that contains statistics for entities in the corpus.

REAnnotatedDocumentView: REDocumentAnnotation View for document entity and event annotations

REEntityStatisticsView: *REProcess* View that contains statistics for entities in REProcess

REProcessAnnotationDocumentsView: REProcess View of all document and the creation of REDocumentAnnotation.

REProcessRelationsResumeStats: REProcess Relations main statistics.

RERelationsViewer: REProcess view of all Relation present in the process.

Configuration

```
<plugin start="true">
    <uid>pt.uminho.anote.aibench.corpora</uid>
    <name>anote corpora</name>
    <version>2.0.0</version>

    <dependencies>
        <dependency uid="pt.uminho.anote.core"/>
        <dependency uid="pt.uminho.anote.datastructures"/>
        <dependency uid="pt.uminho.anote.aibench.utils"/>
    </dependencies>
```

Figure 13 - Anote2CorporaUI AIbench configuration file.

Anote2CuratorUI

Plug-in for manual curation. The main functionalities are the manual creation and correction of annotations, over a publication that was the target of an information extraction process. The user can also start up a document annotation based in clean text.

Operations

OperationManualCuration: Operation for the creation of a Manual Curation Process.

Views

ANoteDocumentView: NERDocumentAnnotation View that allows the creation and edition of annotations for NERDocumentAnnotation.

Configuration

```
<plugin start="true">
    <uid>pt.uminho.anote.aibench.curator</uid>
    <name>anote curator</name>
    <version>2.0.0</version>

    <dependencies>
        <dependency uid="pt.uminho.anote.core"/>
        <dependency uid="pt.uminho.anote.datastructures"/>
        <dependency uid="pt.uminho.anote.aibench.utils"/>
        <dependency uid="pt.uminho.anote.aibench.corpora"/>
    </dependencies>
```

Figure 14 - Anote2CuratorUI AIbench configuration file.

Anote2ResourcesUI

Lexical Resources Plug-in. Allows the management of resources with the possibility to create and edit new resources.

Main classes:

Data-types

Dictionaries: Represents a Dictionary Set

DictionaryAibench: Represents a Dictionary.

LookupTables: Represents a Lookup Table Set.

LookupTableAibench: Represents a Lookup Table.

Ontologies: Represents an Ontology Set.

OntologyAibench: Represents an Ontology.

Resources: Contains information about all lexical resources present in @Note and allows adding a new resource (Dictionary, lookup tables, rules or ontologies).

RulesSet: Represent a Rules Set. Allows Rules Set creation.

RulesAibench: Represent a Rule Set and allows the creation of a new rule and changing rules priority.

Operations

InitResources: Plug-in Starting

ExitResources: Exit Plug-in.

CreateDictionary: Creates a new Dictionary

MergeDictionary: Merging Dictionaries

UploadDicionary: Imports new elements and synonyms, from database flat files, for dictionaries.

UpdateDictionaryBiowareHouse: Imports new elements and synonyms, from local BiowareHouse database, for dictionaries.

AddTermToLookupTable: Adding a new Element for Lookup Table.

CreateLookupTable: Creates a Lookup Table.

LoadLookupTableCSV: Imports data, from CSV files, to a Lookup Table.

MergeLookupTable: Merges Lookup Tables.

SaveLookupTableCSV: Saves Lookup Tables in a CSV file.

CreateOntology: Creates a new Ontology.

CreateRulesSet: Creates a new Rules Set.

MergeRuleSet: Merges Rule Sets.

NewRule: Creates a new Rule Set.

Views

DictionariesView: Dictionaries View, showing Dictionaries available and allowing selection adding dictionary in clipboard.

DictionaryView: DictionaryAibench View showing dictionary information. This view permits management of dictionary elements.

LookupTablesView: LookupTables View that shows available Lookup Tables and allows selection adding LookupTable in clipboard.

LookupTableView: LookupTableAibench View showing LookupTable information.

RulesSetView: RulesSet View that shows RuleSet available and allows selection adding RuleSet in clipboard.

RulesView: RulesAibench View showing Rules information and allowing priority changing.

Configuration

```
<plugin start="true">
    <uid>pt.uminho.anote.resources</uid>
    <name>anote resources</name>
    <version>2.0.0</version>

    <dependencies>
        <dependency uid="pt.uminho.anote.core"/>
        <dependency uid="pt.uminho.anote.datastructures"/>
        <dependency uid="pt.uminho.anote.aibench.utils"/>
    </dependencies>
```

Figure 15 - Anote2resourcesUI AIbench configuration file.

Anote2NEResourcesUI

Plug-in that contains NER processes. This plug-in permits entity document annotation based in lexical resources

Operations

OperationNerAnote: Applies NER Processes to Corpus based in lexical resources.

OperationApplySameNERAnote: Applies NER Processes already done in one Corpus for another.

Configuration

```
<plugin start="true">
    <uid>pt.uminho.anote.aibench.ner</uid>
    <name>anote ner (resources)</name>
    <version>2.0.0</version>

    <dependencies>
        <dependency uid="pt.uminho.anote.core"/>
        <dependency uid="pt.uminho.anote.datastructures"/>
        <dependency uid="pt.uminho.anote.aibench.corpora"/>
        <dependency uid="pt.uminho.anote.aibench.utils"/>
    </dependencies>
```

Figure 16 - Anote2NEResourcesUI AIbench configuration file.

Anote2AIBenchUtils

This plug-in has as its main feature to support all AIBench plug-ins. It includes the lifecycle class that runs the initial setup menu for Anote2, creates and edits configuration settings and

terminates the program. It also includes generic units for supporting graphical interfaces features.

Operations

`ExitOperation`: Exit operation.

`CreateConfigurationsFile: Operation For create @Note configuration File.`

`ChangeProxySettingsOperation: Operation for Changing Configuration proxy.`

`ChangeDBSettingsOperation`: Operation for changing Database access Credentials.

Configuration

```
<plugin start="true">
    <uid>pt.uminho.anote.aibench.utils</uid>
    <name>anote aibench utils</name>
    <version>2.0.0</version>
    <lifecycleclass>pt.uminho.anote.aibench.utils.lifecycle.Lifecycle</lifecycleclass>

    <dependencies>
        <dependency uid="pt.uminho.anote.datastructures"/>
        <dependency uid="pt.uminho.anote.core"/>
    </dependencies>
```

Figure 17 - Anote2AibenchUtils AIbench configuration file

Anote2CorpusLoaders

Plug-in that contains operations for Anote2 corpus creation derived from existent corpora. In this plug-in it is possible to create a corpus for Genia Event, Yapex Protein and @Notev1. With the loading of the corpora it is possible to include annotation schemas in NER or RE Processes.

Operations

CreateCorpusByAIMEDProteinCorpus: To load an AIMED corpus generating a *NERProcess* with protein annotations in documents.

CreateCorpusByAnotev1Corpus: To load an @Notev1 corpus generating a *NERProcess* with entity annotations in documents.

CreateCorpusByGeniaEventCorpus: To load Genia Event corpus generating a *NERProcess* with entity annotations and a *REProcess* with relations annotated in documents.

Configuration

```
<plugin start="true">
    <uid>pt.uminho.anote2.aibench.corpusloaders</uid>
    <name>@Note2 Corpus Loaders (Plug-in)</name>
    <version>1.0.0</version>

    <dependencies>
        <dependency uid="pt.uminho.anote2.core"/>
        <dependency uid="pt.uminho.anote2.datastructures"/>
        <dependency uid="pt.uminho.anote2.aibench.corpora"/>
        <dependency uid="pt.uminho.anote2.aibench.utils"/>
        <dependency uid="pt.uminho.anote2.gate"/>
    </dependencies>
```

Figure 18 - Anote2CorpusLoaders AIBench configuration file

Anote2Gate51

Plug-in that contains a package for connecting with GATE 5.1

```
<plugin start="true">
    <uid>pt.uminho.anote2.gate</uid>
    <name>@Note2 Gate 5.1 (Plug-in)</name>
    <version>1.0.0</version>

    <dependencies>
        <dependency uid="pt.uminho.anote2.core"/>
        <dependency uid="pt.uminho.anote2.datastructures"/>
    </dependencies>
```

Figure 19 - Anote2Gate51 AIBench configuration file

Anote2Gate6

Plug-in that contains a package for connections to GATE 6.

```
<plugin start="true">
    <uid>pt.uminho.anote2.gate.6.1.47</uid>
    <name>@Note2 Gate6.1 (Plug-in)</name>
    <version>6.1.47</version>

    <dependencies>
        <dependency uid="pt.uminho.anote2.core"/>
        <dependency uid="pt.uminho.anote2.datastructures"/>
    </dependencies>
```

Figure20 - Anote2Gate6 AIBench configuration file

Anote2Rel@tioN

Plug-in that contains relation extraction process development. Relation extraction is, currently, based in pre-existent NER Processes with annotations and using Natural Language processing.

Operations

OperationRelationExtraction: Apply the RE Process to Corpus based in Natural Language processing associated with an existent NERProcess for entity identification.

Configuration

```
<plugin start="false">
    <uid>pt.uminho.anote2.aibench.relation</uid>
    <name>@Note2 Rel@tion</name>
    <version>1.0.0</version>


    <dependencies>
        <dependency uid="pt.uminho.anote2.core"/>
        <dependency uid="pt.uminho.anote2.datastructures"/>
        <dependency uid="pt.uminho.anote2.aibench.corpora"/>
        <dependency uid="pt.uminho.anote2.aibench.utils"/>
        <dependency uid="pt.uminho.anote2.gate"/>
    </dependencies>
```

Figure 21 - Anote2Relation AIBench configuration file

Plug-in Dependencies

Plug-in	Dependencies
Anote2Core	
Anote2DataStructures	Anote2Core
Anote2AibenchUtils	Anote2Core Anote2DataStructures
AnoteResourcesUI	Anote2Core Anote2DataStructures Anote2AIbenchUtils
Anote2PubmedRetrievalUI	Anote2Core Anote2DataStructures Anote2AIBenchUtils
Anote2ResourcesUI	Anote2Core Anote2DataStructures Anote2AIbenchUtils
AnoteCorporaUI	Anote2Core Anote2DataStructures Anote2AIbenchUtils
Anote2NEResourcesUI	Anote2Core Anote2DataStructures Anote2Corpora Anote2AIBenchUtils
Anote2CorpusLoaders	Anote2Core Anote2DataStructures Anote2Corpora Anote2AIBenchUtils Anote2Agate51
Anote2Gate51	Anote2Core Anote2DataStructures
Anote2Gate6	Anote2Core Anote2DataStructures
Anote2RelatioN	Anote2Core Anote2DataStructures Anote2Corpora Anote2AIBenchUtils Anote2Agate51