txtDataNLP Framework – Documentation

Table of Contents

[1 Introduction 1](#_Toc31536190)

[2 Central Classes and Concepts 2](#_Toc31536191)

[2.1 TextUnit 2](#_Toc31536192)

[2.2 Word 2](#_Toc31536193)

[2.3 Span 2](#_Toc31536194)

[2.4 Annotation 2](#_Toc31536195)

[2.5 IAnnotationObject 2](#_Toc31536196)

[2.6 AbstractAnnotator 2](#_Toc31536197)

[2.7 AbstractCreator 2](#_Toc31536198)

[2.8 Language 2](#_Toc31536199)

[3 Annotators 2](#_Toc31536200)

[3.1 What is an Annotator? 2](#_Toc31536201)

[3.2 WhitelistAnnotator 2](#_Toc31536202)

[3.3 PosPatternAnnotator 2](#_Toc31536203)

[3.4 SubsumedAnnotationsRemover 2](#_Toc31536204)

[3.5 RecursiveDictionaryAnnotator 2](#_Toc31536205)

[4 Common use cases 2](#_Toc31536206)

[4.1 Creating a simple dictionary-based annotator 3](#_Toc31536207)

[4.2 Creating a more complex annotator based on “POS-Rules” 3](#_Toc31536208)

[4.3 Working with annotations from a third-party Machine Learning model 3](#_Toc31536209)

[4.4 Implementing white- and blacklists on top of a third-party Machine Learning model 3](#_Toc31536210)

[4.5 Building a complex NLP-Pipeline that combines various Machine Learning and/or rule-based approaches 3](#_Toc31536211)

# Introduction

txtDataNLP is a light Java framework for Natural Language Processing (NLP) projects.

Typical use cases include:

* Combining various third-party NLP and Machine Learning (ML) tools into one setup.
* Implementing algorithms that combine output from various NLP/ML modules and return one coherent text analysis—also if their output is contradictory.
* Extraction of features from low-level NLP tools (tokenizers, taggers) for downstream ML models.
* Implementing dictionary-based, rule-based or linguistically motivated approaches to NLP.
* Implementing white- or backlists that often sit on top of ML approaches in order to deal with specific mistakes the ML models make.

txtDataNLP contains no ML algorithms or models itself but is was built to support the use of ML models in NLP and to provide infrastructure to combine various NLP/ML modules, use them together reliably and to fill the gaps that they commonly leave.

# Central Classes and Concepts

## TextUnit

## Word

## Span

## Annotation

## IAnnotationObject

## AbstractAnnotator

## AbstractCreator

## Language

# Annotators

## What is an Annotator?

## WhitelistAnnotator

## PosPatternAnnotator

## SubsumedAnnotationsRemover

## RecursiveDictionaryAnnotator

# Common use cases

## Creating a simple dictionary-based annotator

## Creating a more complex annotator based on “POS-Rules”

## Working with annotations from a third-party Machine Learning model

## Implementing white- and blacklists on top of a third-party Machine Learning model

## Building a complex NLP-Pipeline that combines various Machine Learning and/or rule-based approaches