Package 'stanza'

June 2, 2025

Type Package

Title 'Stanza' - A 'R' NLP Package for Many Human Languages

Version 1.0-3

Description An interface to the 'Python' package 'stanza' <https://stanfordnlp.github.io/stanza/index.html>.
'stanza' is a 'Python' 'NLP' library for many human languages.
It contains support for running various accurate natural language processing tools on 60+ languages.

License GPL-3

Imports checkmate, reticulate

Depends NLP

SystemRequirements R >= 4.0, Python >= 3.8, stanza >= 1.3.0

Encoding UTF-8

NeedsCompilation no

Author Kurt Hornik [aut],
Florian Schwendinger [aut, cre],
Julian Amon [aut]

Maintainer Florian Schwendinger <FlorianSchwendinger@gmx.at>

Repository CRAN

Date/Publication 2025-06-02 08:50:02 UTC

Contents

`conda_install_stanza` *Conda Install Stanza*

Description

Conda Install Stanza

Usage

```
conda_install_stanza(
  envname = "stanza",
  packages = c("python", "stanza"),
  forge = FALSE,
  channel = c("stanfordnlp"),
  conda = "auto",
  ...
)
```

Arguments

`envname`	a character string giving the name or path of the conda environment to be used or created for the installation.
`packages`	a character vector giving the packages to be installed.
`forge`	a logical giving if conda forge should be used for the installation.
`channel`	a character vector giving the conda channels to be used.
`conda`	a character string giving the path to the conda executable.
`...`	additional arguments passed to `conda_install`.

Value

`NULL`

Examples

```
## Not run:
conda_install_stanza()

## End(Not run)
```

entities *Entities*

Description

Entities

Usage

```
entities(x, ...)
```

Arguments

x	an object inheriting from "stanza_document".
...	optional additional arguments, currently not used.

Value

a data.frame with the entities.

is_stanza_initialized *Check if Stanza is Initialized*

Description

Checks if Stanza is initialized.

Usage

```
is_stanza_initialized()
```

Value

TRUE if Stanza is initialized, otherwise FALSE

Examples

```
is_stanza_initialized()
```

multi_word_token *Multi-Word Token*

Description

Multi-Word Token

Usage

```
multi_word_token(x, ...)
```

Arguments

x	an object of
...	optional additional arguments, currently not used.

Value

a data.frame with the multi-word tokens.

stanza_download *Download Models*

Description

Download pretrained NLP models. For more information about the parameters see https://stanfordnlp.github.io/stanza/download_models.html.

Usage

```
stanza_download(
  language = "en",
  model_dir = stanza_options("model_dir"),
  package = "default",
  processors = list(),
  logging_level = "INFO",
  resources_url = stanza_options("resources_url"),
  resources_version = stanza_options("resources_version"),
  model_url = stanza_options("model_url")
)
```

Arguments

language	a character string giving the language (default is "en").
model_dir	path to the directory for storing the for Stanza models (default is "~/stanza_resources").
package	a character string giving the package to be used (default is "default". In this context package refers to a language specific set of models packaged together to a single ".zip" file.
processors	a character string or named list giving the processors to download models for. If a string is provided it should provide the names of the desired processers as comma seperated string, e.g., "tokenize,pos". If a named list is provided, the name should be the processor name and the values the package name, e.g., list(tokenize = "ewt", pos = "ewt").
logging_level	a character string giving the logging level (default is "INFO"), available levels are c('DEBUG', 'INFO', 'WARNING', 'WARN', 'ERROR', 'CRITICAL', 'FATAL').
resources_url	a character string giving the url to the Stanza model resources. The default value is obtained from Python during the initiatlization and can be obtained and changed by using stanza_options.
resources_version	a character string giving the version of the resources. The default value is obtained from Python during the initiatlization and can be obtained and changed by using stanza_options.
model_url	a character string giving the model url. The default value is obtained from Python during the initiatlization and can be obtained and changed by using stanza_options.

Value

NULL

Examples

```
if (stanza_options("testing_level") >= 3L) {
stanza_initialize()
stanza_download("en")
}
```

---

stanza_download_method_code

Select Download Method

Description

Function to obtain the download method code or list all allowed download methods.

Usage

```
stanza_download_method_code(method = NULL)
```

Arguments

method	a character string giving the name of the download method. The case oft he download method name is ignored. If NULL all allowed download methods are shown.

Value

an integer giving the download method code.

Examples

```
if (is_stanza_initialized()) {
  stanza_download_method_code()
  stanza_download_method_code("none")
  stanza_download_method_code("reuse_resources")
  stanza_download_method_code("download_resources")
}
```

---

stanza_initialize *Initialize Stanza*

Description

Initialize the Python binding to stanza.

Usage

```
stanza_initialize(
  python = NULL,
  virtualenv = NULL,
  condaenv = NULL,
  model_dir = NULL,
  resources_url = NULL,
  model_url = NULL
)
```

Arguments

python	a character string giving the path to the Python binary (executeable) to be used. The variable python is passed to reticulate::use_python.
virtualenv	a character string giving the name of the virtual environment, or the path to the virtual environment, to be used. The variable virtualenv is passed to reticulate::use_virtualenv.
condaenv	a character string giving the name of the Conda environment to be used. The variable condaenv is passed to reticulate::use_condaenv.
model_dir	a character sting giving the path to the directory storing the Stanza models.

resources_url	a character string giving the url to the Stanza model resources.
model_url	a character string giving the model url.

Value

NULL

Examples

```
if (stanza_options("testing_level") >= 3L) {
stanza_initialize()
}
```

---

stanza_options *Options*

Description

Allow the user to set and examine options like

Usage

```
stanza_options(option, value, update_python_defaults = FALSE)
```

Arguments

option	any options can be defined, using 'key, value' pairs. If 'value' is missing the current set value is returned for the given 'option'. If both are missing. all set options are returned.
value	the corresponding value to set for the given option.
update_python_defaults	a logical (default is FALSE) controling if the corresponding **stanza** variables should also updated in Python.

Value

- NULL if both arguments option and value are provided.
- The currently set value if the argument value is missing.
- All set options if the argument option is missing.

Examples

```
stanza_options("conda_environment", "stanza")
```

`stanza_pipeline` *NLP Pipeline*

Description

NLP Pipeline

Usage

```
stanza_pipeline(
  language = "en",
  model_dir = stanza_options("model_dir"),
  package = "default",
  processors = list(),
  logging_level = "INFO",
  use_gpu = FALSE,
  download_method = "reuse_resources",
  ...
)
```

Arguments

`language`	a character string giving the language (default is `"en"`).
`model_dir`	path to the directory for storing the for `Stanza` models (default is `"~/stanza_resources"`).
`package`	(default is `"default"`.
`processors`	FIXME: we should define if we want to use comma seperated string or a character vector.
`logging_level`	a character string giving the logging level (default is `"INFO"`), available levels are `c('DEBUG', 'INFO', 'WARNING', 'WARN', 'ERROR', 'CRITICAL', 'FATAL')`.
`use_gpu`	a logical giving if `GPU` or `CPU` should be used (default is `FALSE`).
`download_method`	an integer or character string giving the download method code. If a character string is provided, it is passed to `stanza_download_method_code` to obtain the integer code. Use `stanza_download_method_code` to obtain the code and list all available download methods.
`...`	additional named arguments passed to the stanza pipeline.

Value

a function that can be used to process text.

Examples

```
## Not run:
p <- stanza_pipeline()
doc <- p('R is a programming language for statistical computing.')

## End(Not run)
```

`stanza_version` *Stanza Version*

Description

Obtain the version of the **stanza** Python package.

Usage

```
stanza_version()
```

Value

a character string giving the version of the **stanza** Python package.

Examples

```
stanza_version()
```

`tokens` *Tokens*

Description

Tokens

Usage

```
tokens(x, ...)
```

Arguments

`x`	an object inheriting from `"stanza_document"` or `"stanza_sentence"`.
`...`	optional additional arguments, currently not used.

Value

a data.frame with the tokens.

virtualenv_install_stanza *Install Stanza via Virtual Environment*

Description

Install Stanza via Virtual Environment

Usage

```
virtualenv_install_stanza(
  envname = "stanza",
  packages = "stanza",
  python = NULL,
  ...
)
```

Arguments

envname	a character string giving the name or path of the virtual environment to be used or created for the installation.
packages	a character vector giving the packages to be installed.
python	a string giving the name or path of the python version to be used (e.g., "python3").
...	additional arguments passed to conda_install.

Value

NULL

Examples

```
## Not run:
virtualenv_install_stanza()

## End(Not run)
```

Index