

Package ‘poldis’

July 21, 2024

Type Package

Title Analyse Political Texts

Version 0.1.1

Date 2024-07-17

Maintainer Henrique Sposito <henrique.sposito@graduateinstitute.ch>

Description Wrangle and annotate different types of political texts.
It also introduces Urgency Analysis,
a new method for the analysis of urgency in political texts.

URL <http://henriquesposito.com/poldis/>

BugReports <https://github.com/henriquesposito/poldis/issues>

License MIT + file LICENSE

Imports dplyr, stringr, purrr, stringi, quanteda, spacyr, textstem,
tidyr, stringdist

Suggests rmarkdown, testthat, tesseract, quanteda.textstats, keyATM,
messydates, pdftools, fmsb, ggplot2, tm

RoxygenNote 7.3.1

Encoding UTF-8

LazyData True

Depends R (>= 3.5.0)

NeedsCompilation no

Author Henrique Sposito [cre, aut, ctb] (IHEID,
<<https://orcid.org/0000-0003-3420-6085>>),
James Hollway [ctb] (IHEID, <<https://orcid.org/0000-0002-8361-9647>>),
Jael Tan [ctb] (IHEID, <<https://orcid.org/0000-0002-6234-9764>>)

Repository CRAN

Date/Publication 2024-07-21 08:50:02 UTC

Contents

annotate_text	2
extract_context	3
extract_date	3
extract_locations	4
extract_match	4
extract_names	5
extract_text_similarities	5
extract_title	6
gather_related_terms	7
gather_topics	7
get_urgency	8
read_pdf	9
select_priorities	9
split_text	10
US_News_Conferences_1960_1980	10
Index	11

annotate_text	<i>Annotate text with NLP</i>
---------------	-------------------------------

Description

This function relies on ‘{spacyr}’ NLP parsing to annotate texts.

Usage

```
annotate_text(v, level = "words")
```

Arguments

v	Text vector
level	At which level would you like to parse the text? Options include "words" or "sentences". Defaults to "words".

Value

A data frame with syntax information by words or sentences in text.

extract_context	<i>Extract context for string matches</i>
-----------------	---

Description

A function for getting string matches and the context in which they occur.

Usage

```
extract_context(match, v, level = "sentences", n = 1)
```

Arguments

match	Character string to be matched. For multiple strings, please use " " as a separator.
v	Text vector or annotated data frame.
level	At which text level do you want matches to be returned? Defaults to "sentences". Options are sentences, words, and paragraph.
n	Number of sentences or words matched before and after string match. Defaults to 1. That is, one word or one sentence before, and after, string match. For paragraphs, n is always set to one.

Value

A list of string matches and their context.

Examples

```
extract_context(match = "war|weapons of mass destruction|conflict|NATO|peace",  
               v = US_News_Conferences_1960_1980$text[100],  
               level = "sentences", n = 2)
```

extract_date	<i>Extract dates from text</i>
--------------	--------------------------------

Description

Wrapper function for 'messydates::as_messydates'.

Usage

```
extract_date(v)
```

Arguments

v	Text vector.
---	--------------

Value

A vector of the dates in text.

extract_locations	<i>Extract locations from strings</i>
-------------------	---------------------------------------

Description

Extract locations from strings

Usage

```
extract_locations(v)
```

Arguments

v Text vector.

Details

The function relies on geographical entity detection from NLP models.

Value

A data frame of locations and the number of times they appear.

extract_match	<i>Extract text matches</i>
---------------	-----------------------------

Description

Get texts in which certain "matches" occur.

Usage

```
extract_match(v, match, invert = FALSE, ignore.case = TRUE)
```

Arguments

v Text vector or annotated data frame.

match A regex match for a word(s) or expression. For multiple words, please use "|" to divide them.

invert Do you want texts without certain matches to be returned? By default FALSE.

ignore.case Should case be ignored? By default, TRUE.

Value

A list the same length as text variable.

Examples

```
extract_match(c("This function was created on the 29 September 2021",  
"Today is October 12, 2021"), "October")
```

extract_names	<i>Extract a list of possible names of individuals in texts</i>
---------------	---

Description

Extract a list of possible names of individuals in texts

Usage

```
extract_names(v)
```

Arguments

v A text vector.

Details

The function relies on named entity recognition from NLP models.

Value

A data frame of individual names and the number of times they appear.

extract_text_similarities	<i>Extract similarities and differences in texts/segments</i>
---------------------------	---

Description

Extract similarities and differences in texts/segments

Usage

```
extract_text_similarities(v, comparison = "similarities", method)
```

Arguments

v	Text vector or annotated data frame.
comparison	How would you like to compare texts? Options are "similarities", for comparing similarities, or "differences", for comparing differences. Defaults to "similarities".
method	A method for checking similarities or differences between texts. For similarities, defaults to "correlation" method. Other methods for similarities include "cosine", "jaccard", "ejaccard", "dice", "edice", "simple matching", and "hamann". For differences, defaults to "euclidean". Other methods for differences include "manhattan", "maximum", "canberra", and "minkowski". For more information on each of these methods and what are the implications in selecting a method, please see <code>?quanteda.textstats::textstat_simil()</code> .

Value

A matrix of similarity scores between texts.

extract_title	<i>Extract first sentence from text</i>
---------------	---

Description

A lot of information is contained in the first sentence of a text. In political texts, for example, dates and locations are often contained in the first sentence of the text.

Usage

```
extract_title(v)
```

Arguments

v	Text vector.
---	--------------

Value

A list of the first sentences in text.

Examples

```
extract_title("This is the first sentence. This is the second sentence.")
```

gather_related_terms *Gather terms related to subjects*

Description

Gather terms related to subjects

Usage

```
gather_related_terms(.data, dictionary)
```

Arguments

.data	A data frame, priorities data frame coded using ‘select_priorities()’, or text vector. For data frames, function will search for "text" variable. For priorities data frame function will search for "priorities" variable.
dictionary	The dictionary of 20 major political topics from the Comparative Agendas Project (Jones et al., 2023) is used by default. Users can also declare a custom dictionary as a vector or a list. If users declare a vector, each element is treated as a independent topic. If users declare a list of subjects and related terms, function understands names as topic and words as terms.

Details

This function relies on keyword assisted topic models implemented in the ‘{keyATM}’ package to find related words based on the topics provided and texts in which they appear.

Value

A list of related terms to each of the topics declared in dictionary.

References

Eshima S, Imai K, and Sasaki T. 2024. “Keyword-Assisted Topic Models.” *American Journal of Political Science*, 68(2): 730-750. doi:[10.1111/ajps.12779](https://doi.org/10.1111/ajps.12779)

gather_topics *Gather topic from political discourses*

Description

Gather topic from political discourses

Usage

```
gather_topics(.data, dictionary = "CAP")
```

Arguments

.data	A data frame, priorities data frame coded using ‘select_priorities()’, or text vector. For data frames, function will search for "text" variable. For priorities data frame function will search for "priorities" variable.
dictionary	The dictionary of 20 major political topics from the Comparative Agendas Project (Jones et al., 2023) is used by default. Users can also declare a custom dictionary as a vector or a list. If users declare a vector, each element is treated as a independent topic. If users declare a list of subjects and related terms, function understands names as topic and words as terms.

Value

A list of topics present in each text separated by comma.

Examples

```
gather_topics(US_News_Conferences_1960_1980[1:5, 3])
gather_topics(US_News_Conferences_1960_1980[1:5, 3],
              dictionary = c("military", "development"))
gather_topics(US_News_Conferences_1960_1980[1:5, 3],
              dictionary = list("military" = c("military", "gun", "war"),
                              "development" = c("development", "interest rate", "banks")))
```

get_urgency

Urgency Analysis

Description

Urgency Analysis

Usage

```
get_urgency(.data, normalize = "tokens")
```

Arguments

.data	A data frame, priorities data frame coded using ‘select_priorities()’, or text vector. For data frames, function will search for "text" variable. For priorities data frame function will search for "priorities" variable.
normalize	Would you like urgency scores to be normalized? By default, urgency scores are normalized by "tokens", the number of words in text observation. Users can also declare "none", for no normalization.

Value

A scored data frame for each dimension of urgency.

Examples

```
get_urgency(US_News_Conferences_1960_1980[1:10, 3])
get_urgency(US_News_Conferences_1960_1980[1:10,])
```

read_pdf	<i>Read text from PDFs</i>
----------	----------------------------

Description

Read text from PDFs

Usage

```
read_pdf(path)
```

Arguments

path The path to a PDF file or a folder containing multiple PDFs.

Value

A list of texts.

select_priorities	<i>Select future priorities from political discourses</i>
-------------------	---

Description

Political priorities are statements in which actors express their intent or commitment to take political action in the future.

Usage

```
select_priorities(.data, na.rm = TRUE)
```

Arguments

.data A (annotated) data frame or text vector. For data frames, function will search for "text" variable. For annotated data frames, please declare an annotated data frame at the sentence level.

na.rm Would you like political statements that do not contain a political action to be removed? By default, TRUE.

Value

A data frame with syntax information by sentences and a variable identifying which of these sentences are priorities.

split_text	<i>Split texts</i>
------------	--------------------

Description

Split texts into structured lists of lists according to a split sign.

Usage

```
split_text(v, splitsign = "\\.")
```

Arguments

v	Text vector or annotated data frame.
splitsign	Where do you want to split? By default sentences ("."). This can also be words, signals or other markers you want. For special characters, please use escape sign before (i.e. "\\").

Value

A list of lists the same length as vector.

Examples

```
split_text("This is the first sentence. This is the second sentence.")
```

US_News_Conferences_1960_1980

US News Conferences Data from 1960 to 1980

Description

A dataset containing the news conferences from US presidents from 1960 to 1980. The dataset was gathered from the American Presidency Project website.

Usage

```
data(US_News_Conferences_1960_1980)
```

Format

A data frame with 353 rows and 3 variables: the president, the date, and the full text.

Index

* datasets

US_News_Conferences_1960_1980, [10](#)

annotate_text, [2](#)

extract_context, [3](#)

extract_date, [3](#)

extract_locations, [4](#)

extract_match, [4](#)

extract_names, [5](#)

extract_text_similarities, [5](#)

extract_title, [6](#)

gather_related_terms, [7](#)

gather_topics, [7](#)

get_urgency, [8](#)

read_pdf, [9](#)

select_priorities, [9](#)

split_text, [10](#)

US_News_Conferences_1960_1980, [10](#)