Package: ACEP (via r-universe)

September 7, 2024

Type Package

Title Análisis Computacional de Eventos de Protesta

Version 0.0.3.9005

Description La librería 'ACEP' contiene funciones específicas para desarrollar análisis computacional de eventos de protesta. Asimismo, contiene bases de datos con colecciones de notas sobre protestas y diccionarios de palabras conflictivas. La colección de diccionarios reune diccionarios de diferentes orígenes. The 'ACEP' library contains specific functions to perform computational analysis of protest events. It also contains a database with collections of notes on protests and dictionaries of conflicting words. Collection of dictionaries that brings together dictionaries from different sources.

License MIT + file LICENSE

URL https://github.com/agusnieto77/ACEP, https://agusnieto77.github.io/ACEP/

BugReports https://github.com/agusnieto77/ACEP/issues

Depends R (>= 3.5.0)

Imports graphics, stats, httr, jsonlite, rsyntax, spacyr, udpipe, stringr, reticulate, tidygeocoder

Suggests covr, knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

Encoding UTF-8

Language es

LazyData true

RoxygenNote 7.2.3

VignetteBuilder knitr

Repository https://agusnieto77.r-universe.dev

RemoteUrl https://github.com/agusnieto77/acep

RemoteRef HEAD

RemoteSha 55fe9c7ddc771af44a0b42fe51286d8a86ab37f7

Contents

`acep_bases` *Colección de notas.*

Description

Contiene colecciones de notas de distintos portales noticiosos (una muestra corta). Una segunda colección es de notas del periódico bahiense La Nueva. También tiene resúmenes estadísticos de las bases completas para el desarrollo de los ejemplos de las funciones.

Usage

```
data(acep_bases)
```

Format

Es una lista con 8 objetos.

la_nueva es una url para la descarga de una muestra del corpus de notas de La Nueva Provincia de Bahía Blanca

rev_puerto es una url para la descarga del corpus de notas de la Revista Puerto

rp_procesada es un data frame con indicadores de conflictividad basados en los datos de la Revista Puerto

lc_mdp es una url para la descarga del corpus de notas de La Capital

rp_mdp es una url para la descarga del corpus de notas de la Revista Puerto

ed_neco es una url para la descarga del corpus de notas de Ecos Diarios

ln_bb es una url para la descarga del corpus de notas de La Nueva

ln_arg es una url para la descarga del corpus de notas de La Nación

spacy_postag es un data.frame con una oración procesada con spacyr

titulares es un vector con titulares de notas referidas a conflictos sociales

Source

Revista Puerto

La Nueva

References

Nieto, Agustín 2020 «Intersecciones entre historia digital e historia social: un ejercicio de lectura distante sobre la conflictividad marítima en la historia argentina reciente». Drassana: revista del Museu Maritim (28):122-42. (Revista Drassana)

Examples

```
acep_bases$rp_procesada[1:6, ]
```

`acep_clean` *Limpieza de texto.*

Description

Función que limpia y normaliza las notas/textos.

Usage

```
acep_clean(
  x,
  tolower = TRUE,
  rm_cesp = TRUE,
  rm_emoji = TRUE,
  rm_hashtag = TRUE,
  rm_users = TRUE,
  rm_punt = TRUE,
  rm_num = TRUE,
  rm_url = TRUE,
  rm_meses = TRUE,
  rm_dias = TRUE,
  rm_stopwords = TRUE,
  rm_shortwords = TRUE,
  rm_newline = TRUE,
  rm_whitespace = TRUE,
  other_sw = NULL,
  u = 1
)
```

Arguments

x	vector de textos al que se le aplica la función de limpieza de texto.
tolower	convierte los textos a minúsculas.
rm_cesp	remueve caracteres especiales.
rm_emoji	remueve los emojis.
rm_hashtag	remueve los hashtags.
rm_users	remueve las menciones de usuarixs de redes sociales.
rm_punt	remueve la puntuación.
rm_num	remueve números.
rm_url	remueve las url.
rm_meses	remueve los meses del año.
rm_dias	remueve los dias de la semana.
rm_stopwords	remueve palabras vacías.
rm_shortwords	remueve las palabras cortas.
rm_newline	remueve los saltos de línea.
rm_whitespace	remueve los espacios en blanco.
other_sw	su valor por defecto es NULL, sirve para ampliar el listado de stopwords con un nuevo vector de palabras.
u	umbral de caracteres para la función rm_shortwords.

Value

Si todas las entradas son correctas, la salida sera un vector de textos normalizados.

Examples

```
acep_clean("El SUTEBA fue al paro. Reclaman mejoras salariales.", rm_punt = FALSE)
```

`acep_cleaning` *Limpieza de texto.*

Description

Función que limpia y normaliza las notas/textos.

Usage

```
acep_cleaning(
  x,
  tolower = TRUE,
  rm_cesp = TRUE,
  rm_emoji = TRUE,
  rm_hashtag = TRUE,
  rm_users = TRUE,
  rm_punt = TRUE,
  rm_num = TRUE,
  rm_url = TRUE,
  rm_meses = TRUE,
  rm_dias = TRUE,
  rm_stopwords = TRUE,
  rm_shortwords = TRUE,
  rm_newline = TRUE,
  rm_whitespace = TRUE,
  other_sw = NULL,
  u = 1
)
```

Arguments

`x`	vector de textos al que se le aplica la función de limpieza de texto.
`tolower`	convierte los textos a minúsculas.
`rm_cesp`	remueve caracteres especiales.
`rm_emoji`	remueve los emojis.
`rm_hashtag`	remueve los hashtags.
`rm_users`	remueve las menciones de usuarixs de redes sociales.
`rm_punt`	remueve la puntuación.
`rm_num`	remueve números.
`rm_url`	remueve las url.
`rm_meses`	remueve los meses del año.

rm_dias	remueve los dias de la semana.
rm_stopwords	remueve palabras vacías.
rm_shortwords	remueve las palabras cortas.
rm_newline	remueve los saltos de linea.
rm_whitespace	remueve los espacios en blanco.
other_sw	su valor por defecto es NULL, sirve para ampliar el listado de stopwords con un nuevo vector de palabras.
u	umbral de caracteres para la función rm_shortwords.

Value

Si todas las entradas son correctas, la salida sera un vector de textos normalizados.

Examples

```
acep_cleaning("El SUTEBA fue al paro. Reclaman mejoras salariales.", rm_stopword = FALSE)
acep_cleaning("El SUTEBA fue al paro. Reclaman mejoras salariales.", rm_stopword = TRUE)
```

---

acep_context *Función para extraer contexto de palabras o frases.*

Description

Función que devuelve un data.frame con el contexto de una o más palabras o frases según una ventana de palabras hacia las izquierda y otra ventana de palabras hacia la derecha.

Usage

```
acep_context(texto, clave, izq = 1, der = 1, ci = "\\b", cd = "\\S*")
```

Arguments

texto	vector con los textos a procesar.
clave	vector de palabras clave a contextualizar.
izq	número de palabras de la ventana hacia la izquierda.
der	número de palabras de la ventana hacia la derecha.
ci	expresión regular a la izquierda de la palabra clave.
cd	expresión regular a la derecha de la palabra clave.

Value

Si todas las entradas son correctas, la salida sera un data frame con el id de los textos procesado y el contexto de las palabras y/o frases entradas.

Examples

```
texto <- "El SOIP para por aumento de salarios"
texto_context <- acep_context(texto = texto, clave = "para")
texto_context
```

---

acep_count *Frecuencia de menciones de palabras.*

Description

Reemplaza a la función 'acep_men' que cuenta la frecuencia de menciones de palabras que refieren a conflictos en cada una de las notas/textos.

Usage

```
acep_count(texto, dic)
```

Arguments

texto	vector de textos al que se le aplica la función de conteo de la frecuencia de menciones de palabras del diccionario.
dic	vector de palabras del diccionario utilizado.

Value

Si todas las entradas son correctas, la salida sera un vector con una frecuencia de palabras de un diccionario.

Examples

```
df <- data.frame(texto = c("El SUTEBA fue al paro. Reclaman mejoras salariales.",
"El SOIP lleva adelante un plan de lucha con paros y piquetes."))
diccionario <- c("paro", "lucha", "piquetes")
df$detect <- acep_men(df$texto, diccionario)
df
```

acep_db *Frecuencia, menciones e intensidad.*

Description

Función que usa las funciones acep_frec, acep_men y acep_int y devuelve una tabla con tres columnas nuevas: numero de palabras, número de menciones del diccionario, indice de intensidad.

Usage

```
acep_db(db, t, d, n)
```

Arguments

db	data frame con los textos a procesar.
t	columna de data frame que contiene el vector de textos a procesar.
d	diccionario en formato vector.
n	cantidad de decimales del indice de intensidad.

Value

Si todas las entradas son correctas, la salida sera una base de datos en formato tabular con tres nuevas variables.

Examples

```
df <- data.frame(texto = c("El SUTEBA fue al paro. Reclaman mejoras salariales.",
"El SOIP lleva adelante un plan de lucha con paros y piquetes."))
diccionario <- c("paro", "lucha", "piquetes")
acep_db(df, df$texto, diccionario, 4)
```

acep_detect *Detección de menciones de palabras.*

Description

Función que detecta de menciones de palabras que refieren a conflictos en cada una de las notas/textos.

Usage

```
acep_detect(x, y, u = 1, tolower = TRUE)
```

Arguments

x	vector de textos al que se le aplica la función de detección de menciones de palabras del diccionario.
y	vector de palabras del diccionario utilizado.
u	umbral para atribuir valor positivo a la detección de las menciones.
tolower	convierte los textos a minúsculas.

Value

Si todas las entradas son correctas, la salida sera un vector numérico.

Examples

```
df <- data.frame(texto = c("El SUTEBA fue al paro. Reclaman mejoras salariales.",
"El SOIP lleva adelante un plan de lucha con paros y piquetes."))
diccionario <- c("paro", "lucha", "piquetes")
df$detect <- acep_detect(df$texto, diccionario)
df
```

---

acep_diccionarios *Colección de diccionarios.*

Description

Colección de diccionarios que reúne diccionarios de diferentes orígenes. El diccionario dicc_confl_acep fueron construidos en el marco del Observatorio de Conflictividad de la UNMdP. Los diccionarios dicc_confl_gp y dicc_viol_gp fueron extraídos de Albrieu y Palazzo (2020).

Usage

```
data(acep_diccionarios)
```

Format

Es un objeto de clase 'list' con 3 componentes.

dicc_confl_gp es un vector con palabras de un diccionario de términos que refieren a conflictos

dicc_viol_gp es un vector con palabras de un diccionario de términos que refieren a violencia

dicc_confl_sismos es un vector con palabras de un diccionario de términos que refieren a conflictos

Source

Revista Puerto

La Nueva

References

Albrieu, Ramiro y Gabriel Palazzo 2020 «Categorización de conflictos sociales en el ámbito de los recursos naturales: un estudio de las actividades extractivas mediante la minería de textos». Revista CEPAL (131):29-59. (Revista CEPAL)

Laitano, Guillermina y Agustín Nieto «Análisis computacional de la conflictividad laboral en Mar del Plata durante el gobierno de Cambiemos». Ponencia presentado en VI Workshop - Los conflictos laborales en la Argentina del siglo XX y XXI: un abordaje interdisciplinario de conceptos, problemas y escalas de análisis, Tandil, 2021.

Examples

```
diccionario <- acep_load_base(acep_diccionarios$dicc_viol_gp)
diccionario
```

`acep_extract` *Función para buscar y extraer palabras en un texto.*

Description

Esta función busca palabras clave en un texto y extrae los resultados en un formato especifico.

Usage

```
acep_extract(texto, dic, sep = "; ", izq = "\\b\\w*", der = "\\w*\\b")
```

Arguments

`texto`	El texto en el que se buscaran las palabras clave.
`dic`	Un vector con las palabras clave a buscar.
`sep`	El separador utilizado para concatenar las palabras clave encontradas (por defecto: " | ").
`izq`	expresión regular para incorporar otros caracteres a la izquierda de los términos del vector 'dic'.
`der`	expresión regular para incorporar otros caracteres a la derecha de los términos del vector 'dic'.

Value

Si todas las entradas son correctas, la salida sera un vector de tipo caracter. procesado y el contexto de las palabras y/o frases entradas.

Examples

```
texto <- "Los obreros del pescado, en el marco de una huelga y
realizaron una manifestación con piquete en el puerto de la ciudad."
dicc <- c("huel", "manif", "piq")
acep_extract(texto, dicc)
```

acep_frec *Frecuencia de palabras totales.*

Description

Función que cuenta la frecuencia de palabras totales en cada una de las notas/textos.

Usage

```
acep_frec(x)
```

Arguments

x	vector de textos al que se le aplica la función de conteo de la frecuencia de palabras.

Value

Si todas las entradas son correctas, la salida sera un vector con una frecuencia de palabras.

Examples

```
acep_frec("El SUTEBA fue al paro. Reclaman mejoras salariales.")
```

acep_gpt *Función para interactuar con los modelos de OpenAI.*

Description

Función para interactuar con los modelos de OpenAI desde las claves API secretas y pagas.

Usage

```
acep_gpt(
  texto,
  instrucciones,
  gpt_api = Sys.getenv("OPENAI_API_KEY"),
  modelo = "gpt-3.5-turbo-0125"
)
```

Arguments

texto	texto a procesar.
instrucciones	indicaciones de contexto que se le proporcionan al modelo de lenguaje GPT para influir en su generación de texto.
gpt_api	clave API secreta.
modelo	modelo de lenguaje GPT de OpenAI.

Value

Si todas las entradas son correctas, la salida sera un JSON con la información solicitada.

Examples

```
## Not run:
texto <- "El SOIP declaro la huelga por aumento de salarios."
pregunta <- "¿Cuál es el sujeto de la acción? Ejemplo de respuesta: 'CGT'"
texto_gpt <- acep_gpt(texto = texto, instrucciones = pregunta)
cat(texto_gpt)

## End(Not run)
```

`acep_int` *Índice de intensidad.*

Description

Función que elabora un indice de intensidad en base a la relación entre palabras totales y palabras del diccionario presentes en el texto.

Usage

```
acep_int(pc, pt, decimales = 4)
```

Arguments

`pc`	vector numérico con la frecuencia de palabras conflictivas presentes en cada texto.
`pt`	vector de palabras totales en cada texto.
`decimales`	cantidad de decimales, por defecto tiene 4 pero se puede modificar.

Value

Si todas las entradas son correctas, la salida sera un vector numérico.

Examples

```
conflictos <- c(1, 5, 0, 3, 7)
palabras <- c(4, 11, 12, 9, 34)
acep_int(conflictos, palabras, 3)
```

acep_load_base *Carga los corpus y las bases creadas por el Observatorio.*

Description

Función para cargar bases de datos disponibles online. Por ahora están disponibles las siguientes bases: Revista Puerto 'rp_mdp'; La Nueva 'ln_bb', La Capital 'lc_mdp', Ecos Diarios 'ed_neco', La Nación 'ln_arg'

Usage

```
acep_load_base(tag)
```

Arguments

tag	etiqueta identificatoria del data frame a cargar: acep_bases$rp_mdp, acep_bases$ln_bb, acep_bases$lc_mdp, acep_bases$ed_neco, acep_bases$ln_arg

Value

Si todas las entradas son correctas, la salida sera una base de datos en formato tabular con un corpus de notas.

Examples

```
bd_sismos <- acep_bases$rev_puerto
head(acep_load_base(tag = bd_sismos))
```

acep_may *Convierte caracteres a mayusculas.*

Description

Esta función toma un vector de texto y convierte todas las letras minúsculas en mayúsculas, manteniendo el resto de los caracteres sin cambios.

Usage

```
acep_may(x)
```

Arguments

x	es un vector de texto (caracteres) que se desea convertir a mayúsculas.

Value

Devuelve un nuevo vector con todas las letras en mayúsculas.

Examples

```
vector_texto <- c("soip", "cGt", "Sutna")
acep_may(vector_texto)
vector_numeros <- c(1, 2, 3, 4, 5)
acep_may(vector_numeros)
vector_mezclado <- c("sutna", 123, "Ate")
acep_may(vector_mezclado)
```

---

acep_men *Frecuencia de menciones de palabras.*

Description

Función que cuenta la frecuencia de menciones de palabras que refieren a conflictos en cada una de las notas/textos.

Usage

```
acep_men(x, y, tolower = TRUE)
```

Arguments

x	vector de textos al que se le aplica la función de conteo de la frecuencia de menciones de palabras del diccionario.
y	vector de palabras del diccionario utilizado.
tolower	convierte los textos a minúsculas.

Value

Si todas las entradas son correctas, la salida sera un vector con una frecuencia de palabras de un diccionario.

Examples

```
df <- data.frame(texto = c("El SUTEBA fue al paro.
Reclaman mejoras salariales.",
"El SOIP lleva adelante un plan de lucha con paros y piquetes."))
diccionario <- c("paro", "lucha", "piquetes")
df$detect <- acep_men(df$texto, diccionario)
df
```

acep_min *Convierte caracteres a minúsculas.*

Description

Esta función toma un vector de texto y convierte todas las letras mayusculas en minúsculas, manteniendo el resto de los caracteres sin cambios.

Usage

```
acep_min(x)
```

Arguments

x	Un vector de texto (caracteres) que se desea convertir a minúsculas.

Value

Devuelve un nuevo vector con todas las letras en minúsculas.

Examples

```
vector_texto <- c("SUTEBA", "Sindicato", "PEN")
acep_min(vector_texto)
vector_numeros <- c(1, 2, 3, 4, 5)
acep_min(vector_numeros)
vector_mezclado <- c("Soip", 123, "CGT")
acep_min(vector_mezclado)
```

acep_plot_rst *Resumen visual de la serie temporal de los indices de conflictividad.*

Description

Función que devuelve un panel visual de cuatro gráficos de barras con variables proxy de los indices de conflictividad agrupados por segmento de tiempo.

Usage

```
acep_plot_rst(datos, tagx = "horizontal")
```

Arguments

datos	data frame con datos procesados.
tagx	orientación de las etiquetas del eje x ('horizontal' | 'vertical').

Value

Si todas las entradas son correctas, la salida sera una imagen de cuatro paneles.

Examples

```
datos <- acep_bases$rp_procesada
fecha <- datos$fecha
n_palabras <- datos$n_palabras
conflictos <- datos$conflictos
datos_procesados_anio <- acep_rst(datos,
fecha, n_palabras, conflictos, st = 'anio')
acep_plot_rst(datos_procesados_anio, tagx = 'vertical')
```

`acep_plot_st` *Gráfico de barras de la serie temporal de indices de conflictividad.*

Description

Función que devuelve un gráfico de barras con la serie temporal de indices de conflictividad por dia, mes o anio.

Usage

```
acep_plot_st(
  x,
  y,
  t = "",
  ejex = "",
  ejey = "",
  etiquetax = "horizontal",
  color = "mint"
)
```

Arguments

`x`	vector de valores del eje x (por ejemplo, fechas).
`y`	vector de valores numéricos del eje y (por ejemplo, menciones).
`t`	titulo del gráfico.
`ejex`	nombre del eje x.
`ejey`	nombre del eje y.
`etiquetax`	orientación de las etiquetas del eje x ('horizontal' | 'vertical').
`color`	color de las barras.

Value

Si todas las entradas son correctas, la salida sera una imagen de un panel.

Examples

```
datos <- acep_bases$rp_procesada
fecha <- datos$fecha
n_palabras <- datos$n_palabras
conflictos <- datos$conflictos
dpa <- acep_rst(datos,
fecha, n_palabras, conflictos, st = 'anio')
acep_plot_st(
dpa$st, dpa$frecm,
t = 'Evoluci\u00f3n de la conflictividad en el sector pesquero argentino',
ejex = 'A\u00f1os analizados',
ejey = 'Menciones de t\u00e9rminos del diccionario de conflictos',
etiquetax = 'horizontal')
```

---

acep_postag *Función para etiquetado POS, lematización, tokenización, extracción de entidades.*

Description

Función que devuelve seis objetos data.frame con etiquetado POS (modelo spacyr) para su posterior procesamiento con la función acep_postag.

Usage

```
acep_postag(
  texto,
  core = "es_core_news_lg",
  bajar_core = TRUE,
  inst_spacy = FALSE,
  inst_miniconda = FALSE,
  inst_reticulate = FALSE
)
```

Arguments

texto	vector con los textos a procesar.
core	idioma del modelo de etiquetado POS del paquete spacyr.
bajar_core	parámetro booleano que define si bajar o no el modelo de etiquetado POS.
inst_spacy	parámetro booleano que define si instalar o no spacy (python).
inst_miniconda	parámetro booleano que define si instalar o no miniconda. Miniconda es un instalador mínimo gratuito para conda. Es una pequeña versión de arranque de Anaconda que incluye solo conda, Python, los paquetes de los que dependen y una pequeña cantidad de otros paquetes útiles, incluidos pip, zlib y algunos otros.
inst_reticulate	parámetro booleano que define si instalar o no el paquete reticulate.

Value

Si todas las entradas son correctas, la salida sera una lista con seis bases de datos en formato tabular.

Source

Dependencias Universales para taggeo POS

Sobre el modelo spaCy de Python para R

Sobre el paquete tidygeocoder

Examples

```
## Not run:
texto <- "En Mar del Plata el SOIP declara la huelga en demanda de aumento salarial."
texto_postag <- acep_postag(texto)
texto_postag$texto_tag[ , c(1:8)]

## End(Not run)
```

`acep_prompt_gpt` *Colección de instrucciones para GPT.*

Description

Colección de instrucciones para interactuar con los modelos de OpenAI. Las instrucciones fueron testeadas en el marco de las tareas que realizamos en el Observatorio de Conflictividad Social de la Universidad Nacional de Mar del Plata.

Usage

```
data(acep_prompt_gpt)
```

Format

Es un objeto de clase 'list' con 4 componentes.

instruccion_breve_sao_es es un texto en castellano con instrucciones breves para extraer eventos de protesta y codificarlos con las siguientes claves: 'fecha', 'sujeto', 'accion', 'objeto', 'lugar'.

instruccion_larga_sao_es es un texto en castellano con instrucciones largas para extraer eventos de protesta y codificarlos con las siguientes claves: 'id', 'cronica', 'fecha', 'sujeto', 'organizacion', 'participacion', 'accion', 'objeto', 'lugar'.

instruccion_breve_sao_en es un texto en inglés con instrucciones breves para extraer eventos de protesta y codificarlos con las siguientes claves: 'date', 'subject', 'action', 'object', 'place'.

instruccion_larga_sao_en es un texto en inglés con instrucciones largas para extraer eventos de protesta y codificarlos con las siguientes claves: 'id', 'chronicle', 'date', 'subject', 'organization', 'participation', 'action', 'object', 'place'.

Examples

```
prompt01 <- acep_prompt_gpt$instrucciones_gpt
prompt01
```

`acep_rs` *Cadenas de caracteres para limpiar y normalizar textos.*

Description

Cadenas de caracteres y expresiones regulares para limpiar y normalizar textos.

Usage

```
data(acep_rs)
```

Format

Son cadenas de caracteres.

sw1 es un string de palabras vacias.

sw1 es un string de palabras vacias.

dias es un string de dias.

meses es un string de meses.

emoji es un string con expresiones regulares para emojis.

sintildes es un string de letras sin tildes.

tildes es un string de letras con tildes.

punt es un string de puntuación.

num es una expresión regular para números.

hashtag es una expresión regular para hashtag.

espacios es una expresión regular para espacios.

saltos es una expresión regular para saltos de línea.

url es una expresión regular para urls.

users es una expresión regular para usuarixs.

Examples

```
print(acep_rs)
```

acep_rst *Serie temporal de índices de conflictividad.*

Description

Función que devuelve los indices de conflictividad agrupados por segmento de tiempo: 'dia', 'mes', 'anio'.

Usage

```
acep_rst(datos, fecha, frecp, frecm, st = "mes", u = 2, d = 4)
```

Arguments

datos	data frame con las variables 'fecha' (en formato Date), 'n_palabras' (numérica), conflictos' (numérica), 'intensidad' (numérica).
fecha	columna de data frame que contiene el vector de fechas en formato date.
frecp	columna de data frame que contiene el vector de frecuencia de palabras por texto.
frecm	columna de data frame que contiene el vector de menciones del diccionario por texto.
st	parámetro para establecer el segmento temporal a ser agrupado: anio, mes, día.
u	umbral de menciones para contabilizar una nota como nota que refiere a un conflicto.
d	cantidad de decimales, por defecto tiene 4 pero se puede modificar.

Value

Si todas las entradas son correctas, la salida sera una base de datos en formato tabular con nuevas variables.

Examples

```
datos <- acep_bases$rp_procesada
fecha <- datos$fecha
n_palabras <- datos$n_palabras
conflictos <- datos$conflictos
datos_procesados_anio <- acep_rst(datos,
fecha, n_palabras, conflictos, st = 'anio', u = 4)
datos_procesados_mes <- acep_rst(datos,
fecha, n_palabras, conflictos)
datos_procesados_dia <- acep_rst(datos,
fecha, n_palabras, conflictos, st = 'dia', d = 3)
head(datos_procesados_anio)
head(datos_procesados_mes)
head(datos_procesados_dia)
```

acep_sst *Serie temporal de índices de conflictividad.*

Description

Función que devuelve los indices de conflictividad agrupados por segmento de tiempo: 'dia', 'mes', 'anio'. Esta función viene a reemplazar a acep_rst. Simplifica los parámetros.

Usage

```
acep_sst(datos, st = "mes", u = 2, d = 4)
```

Arguments

datos	data frame con las variables 'fecha' (en formato Date), 'n_palabras' (numérica), conflictos' (numérica), 'intensidad' (numérica). Las ultimas tres se pueden construir en un solo paso con la función 'acep_db' o en tres pasos con las funciones 'acep_frec', 'acep_men', 'acep_int'.
st	parámetro para establecer el segmento temporal a ser agrupado: 'anio', 'mes', 'dia'.
u	umbral de menciones para contabilizar una nota como nota que refiere a un conflicto, por defecto tiene 2 pero se puede modificar.
d	cantidad de decimales, por defecto tiene 4 pero se puede modificar.

Value

Si todas las entradas son correctas, la salida sera una base de datos en formato tabular con nuevas variables.

Examples

```
datos <- acep_bases$rp_procesada
head(datos)
datos_procesados_anio <- acep_sst(datos, st='anio', u=4)
datos_procesados_mes <- acep_sst(datos)
datos_procesados_dia <- acep_sst(datos, st ='dia', d=3)
head(datos_procesados_anio)
head(datos_procesados_mes)
head(datos_procesados_dia)
```

acep_svo *Función para extraer tripletes SVO (Sujeto-Verbo-Objeto).*

Description

Función que devuelve seis objetos data.frame con etiquetado POS (modelo spacyr) y relaciones sintácticas (modelo rsyntax) que permiten reconstruir estructuras sintácticas como SVO y Sujeto-Predicado. Una vez seleccionadas las notas periodísticas referidas a conflictos, esta función permite extraer sujetos de la protesta, acción realizada y objeto(s) de la acción. También devuelve entidades nombradas (NER).

Usage

```
acep_svo(acep_tokenindex, prof_s = 3, prof_o = 3, u = 1)
```

Arguments

acep_tokenindex	data.frame con el etiquetado POS y las relaciones de dependencia generado con la función acep_postag.
prof_s	es un numero entero positivo que determina la profundidad a la que se buscan las relaciones dentro del sujeto. Este parámetro se hereda del la función children() del paquete rsyntax. Se recomienda no superar el valor 2.
prof_o	es un numero entero positivo que determina la profundidad a la que se buscan las relaciones dentro del objeto. Este parámetro se hereda del la función children() del paquete rsyntax. Se recomienda no superar el valor 2.
u	numero entero que indica el umbral de palabras del objeto en la reconstrucción SVO.

Value

Si todas las entradas son correctas, la salida sera una lista con tres bases de datos en formato tabular.

Source

Dependencias Universales para taggeo POS

Sobre el paquete rsyntax

References

Welbers, K., Atteveldt, W. van, & Kleinnijenhuis, J. 2021. Extracting semantic relations using syntax: An R package for querying and reshaping dependency trees. Computational Communication Research, 3-2, 1-16. (link al articulo)

Examples

```
## Not run:
acep_svo(acep_bases$spacy_postag)

## End(Not run)
```

`acep_token` *Tokenizador.*

Description

Función que tokeniza las notas/textos.

Usage

```
acep_token(x, tolower = TRUE, cleaning = TRUE)
```

Arguments

`x`	vector de textos al que se le aplica la función de tokenización.
`tolower`	convierte los textos a minúsculas.
`cleaning`	hace una limpieza de los textos.

Value

Si todas las entradas son correctas, la salida será un data.frame con las palabras tokenizadas.

Examples

```
acep_token("Huelga de obreros del pescado en el puerto")
```

`acep_token_plot` *Gráfico de barras de palabras más recurrentes en un corpus.*

Description

Función que devuelve un gráfico de barras con las palabras mas recurrentes en un corpus textual.

Usage

```
acep_token_plot(x, u = 10, frec = TRUE)
```

Arguments

x	vector de palabras tokenizadas.
u	numero de corte para el top de palabras mas frecuentes.
frec	parámetro para determinar si los valores se visualizaran como frecuencia absoluta o relativa.

Value

Si todas las entradas son correctas, la salida sera un gráfico de barras.

Examples

```
tokens <- c(rep("paro",15), rep("piquete",25), rep("corte",20), rep("manifestación",10),
rep("bloqueo",5), rep("alerta",16), rep("ciudad",12), rep("sindicato",11), rep("paritaria",14),
rep("huelga",14), rep("escrache",15))
acep_token_plot(tokens)
```

---

acep_token_table *Tabla de frecuencia de palabras tokenizadas.*

Description

Función que cuenta la frecuencia de palabras tokenizadas.

Usage

```
acep_token_table(x, u = 10)
```

Arguments

x	vector de palabras tokenizadas.
u	número de corte para el top de palabras más frecuentes.

Value

Si todas las entradas son correctas, la salida sera una tabla con la frecuencia relativa y absoluta de palabras tokenizadas.

Examples

```
tokens <- c(rep("paro",15), rep("piquete",25), rep("corte",20), rep("manifestación",10),
rep("bloqueo",5), rep("alerta",16), rep("ciudad",12), rep("sindicato",11), rep("paritaria",14),
rep("huelga",14), rep("escrache",15))
acep_token_table(tokens)
```

acep_upos *Función para etiquetado POS, lematización, tokenización.*

Description

Función que devuelve un marco de datos objetos con etiquetado POS (modelo udpipe) para su posterior procesamiento con la función acep_postag.

Usage

```
acep_upos(texto, modelo = "spanish")
```

Arguments

texto	vector con los textos a procesar.
modelo	idioma del modelo de etiquetado POS del paquete udpipe.

Value

Si todas las entradas son correctas, la salida sera un marco de datos con 17 variables.

Source

Dependencias Universales para taggeo POS

Sobre el modelo UDPipe

Sobre el paquete rsyntax

References

Welbers, K., Atteveldt, W. van, & Kleinnijenhuis, J. 2021. Extracting semantic relations using syntax: An R package for querying and reshaping dependency trees. Computational Communication Research, 3-2, 1-16. (link al articulo)

Examples

```
## Not run:
texto <- "El SOIP declara la huelga en demanda de aumento salarial."
acep_upos(texto)

## End(Not run)
```

Index