Package 'CEDARS'

January 30, 2021

Type Package

Title Simple and Efficient Pipeline for Electronic Health Record Annotation

Description Streamlined annotation pipeline for collection and aggregation of time-to-event data in retrospective clinical studies. 'CEDARS' aims to systematize and accelerate the review of electronic health record (EHR) corpora. It accomplishes those goals by deploying natural language processing as a tool to assist detection and characterization of clinical events by human abstractors. The online user manual presents the necessary steps to install 'CEDARS', process EHR corpora and obtain clinical event dates: <https://cedars.io>.

Version 1.84

Imports fastmatch,
jsonlite,
mongolite,
parallel,
readr,
shiny,
udpipe,
utils

License GPL-3

URL `https://cedars.io` (main)

`https://github.com/simon-hans/CEDARS` (devel)

BugReports `https://github.com/simon-hans/CEDARS/issues`

Depends R (>= 3.5.0)

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1

Language en-US

Encoding UTF-8

LazyData true

R topics documented:

`add_end_user` *Add a CEDARS End User*

Description

Adds an end user. Password must be at least 8 characters in length.

Usage

```
add_end_user(
  uri_fun,
  user,
  password,
  host,
  port,
  database,
  end_user,
  end_user_password
)
```

Arguments

`uri_fun`	Uniform resource identifier (URI) string generating function for MongoDB credentials.
`user`	MongoDB user name.
`password`	MongoDB user password.
`host`	MongoDB host server.
`port`	MongoDB port.
`database`	MongoDB database name.
`end_user`	CEDARS end user name.
`end_user_password`	CEDARS end user password.

Examples

```
## Not run:
add_end_user(uri_fun = mongo_uri_standard, user = 'John', password = 'db_password_1234',
host = 'server1234', port = NA, database = 'TEST_PROJECT', end_user = 'Mike',
end_user_password = 'user_pw_5678')

## End(Not run)
```

`automatic_NLP_processor` *Process NLP Annotations on the Current Patient Cohort*

Description

Accepts a list of patient ID's or alternatively can perform NLP annotations on all available patients in the database.

Usage

```
automatic_NLP_processor(
  patient_vect = NA,
  text_format = "latin1",
  nlp_engine = "udpipe",
  uri_fun = mongo_uri_standard,
  user,
  password,
  host,
  port,
  database,
  max_n_grams_length = 7,
  negex_depth = 6,
  select_cores = NA,
  URL = NA
)
```

Arguments

`patient_vect`	Vector of patient ID's. Default is NA, in which case all available patient records will undergo NLP annotation.
`text_format`	Text format for NLP engine.
`nlp_engine`	Which NLP engine should be used? UDPipe is the only one supported for now.
`uri_fun`	Uniform resource identifier (URI) string generating function for MongoDB credentials.
`user`	MongoDB user name.
`password`	MongoDB user password.
`host`	MongoDB host server.
`port`	MongoDB port.
`database`	MongoDB database name.
`max_n_grams_length`	Maximum length of tokens for matching with UMLS concept unique identifiers (CUI's). Shorter values will result in faster processing. If 0 is chosen, UMLS CUI tags will not be provided.
`negex_depth`	Maximum distance between negation item and token to negate. Shorter distances will result in decreased sensitivity but increased specificity for negation.
`select_cores`	How many CPU cores should be used for parallel processing? Max allowed is total number of cores minus one. If 1 is entered, parallel processing will not be used.
`URL`	UDPipe model URL.

Examples

```
## Not run:
automatic_NLP_processor(patient_vect = NA, text_format = 'latin1', nlp_engine = 'udpipe',
URL = 'models/english-ewt-ud-2.4-190531.udpipe', uri_fun = mongo_uri_standard, user = 'John',
password = 'db_password_1234', host = 'server1234', port = NA, database = 'TEST_PROJECT',
max_n_grams_length = 7, negex_depth = 6, select_cores = 1)

## End(Not run)
```

`create_project` *Create a New CEDARS Project*

Description

Creates a new MongoDB database and collections needed for a CEDARS annotation project. The MongoDB account used must have sufficient privileges.

Usage

```
create_project(
  uri_fun,
  user,
  password,
  host,
  port,
  database,
  project_name,
  investigator_name
)
```

Arguments

`uri_fun`	Uniform resource identifier (URI) string generating function for MongoDB credentials.
`user`	MongoDB user name.
`password`	MongoDB user password.
`host`	MongoDB host server.
`port`	MongoDB port.
`database`	MongoDB database name.
`project_name`	Research or QA project name.
`investigator_name`	Investigator name.

Examples

```
# The code below creates an instance of CEDARS project on a public test MongoDB cluster, populated
# with fictitious EHR corpora.

# MongoDB credentials
db_user_name <- "testUser"
db_user_pw <- "testPW"
db_host <- "cedars.yvjp6.mongodb.net"
db_port <- NA

# Using standard MongoDB URL format
uri_fun <- mongo_uri_standard

# Name for MongoDB database which will contain the CEDARS project
# In this case we generate a random name
mongo_database <- find_project_name()

# We create the database and all required collections on a test cluster
create_project(uri_fun, db_user_name, db_user_pw, db_host, db_port, mongo_database,
"CEDARS Example Project", "Dr Smith")

# Adding one CEDARS end user
add_end_user(uri_fun, db_user_name, db_user_pw, db_host, db_port, mongo_database, "John",
"strongpassword")

## Not run:
```

```
# Negex is included with CEDARS and required for assessment of negation
negex_upload(uri_fun, db_user_name, db_user_pw, db_host, db_port, mongo_database)

## End(Not run)

# Uploading the small simulated collection of EHR corpora
upload_notes(uri_fun, db_user_name, db_user_pw, db_host, db_port, mongo_database,
simulated_patients)

# This is a simple query which will report all sentences with a word starting in
# "bleed" or "hem", or an exact match for "bled"
search_query <- "bleed* OR hem* OR bled"
use_negation <- TRUE
hide_duplicates <- TRUE
skip_after_event <- TRUE
save_query(uri_fun, db_user_name, db_user_pw, db_host, db_port, mongo_database, search_query,
use_negation, hide_duplicates, skip_after_event)

## Not run:

# Running the NLP annotations on EHR corpora
# We are only using one core, for large datasets parallel processing is faster
automatic_NLP_processor(NA, "latin1", "udpipe", uri_fun, db_user_name, db_user_pw,
db_host, db_port, mongo_database, max_n_grams_length = 0, negex_depth = 6, select_cores = 1)

# Pre-searching based on query
# This is optional but will speed-up the interface
pre_search(patient_vect = NA, uri_fun, db_user_name, db_user_pw, db_host, db_port, mongo_database)

# Start the CEDARS GUI locally
# Your user name is "John", password is "strongpassword"
# Once you have entered those credentials, click on button "ENTER NEW DATE" and CEDARS will
# seek the first record to annotate
# Try out the interface, adjudicating sentences, entering event dates, comments, moving
# between sentences and searching for records
# Once you have entered some data, close the GUI
start_local(db_user_name, db_user_pw, db_host, db_port, mongo_database)

# Obtaining events and info associated with data entry
# The annotations entered in the GUI are now available in this dataframe
event_output <- download_events(uri_fun, db_user_name, db_user_pw, db_host, db_port, mongo_database)

## End(Not run)

# Remove project from MongoDB
terminate_project(uri_fun, db_user_name, db_user_pw, db_host, db_port, mongo_database, fast=TRUE)
```

`delete_end_user` *Delete a CEDARS End USer*

Description

Deletes one end user and associated password.

Usage

```
delete_end_user(uri_fun, user, password, host, port, database, end_user)
```

Arguments

`uri_fun`	Uniform resource identifier (URI) string generating function for MongoDB credentials.
`user`	MongoDB user name.
`password`	MongoDB user password.
`host`	MongoDB host server.
`port`	MongoDB port.
`database`	MongoDB database name.
`end_user`	CEDARS end user name.

Examples

```
## Not run:
delete_end_user(uri_fun = mongo_uri_standard, user = 'John', password = 'db_password_1234',
host = 'server1234', port = NA, database = 'TEST_PROJECT', end_user = 'Mike')

## End(Not run)
```

`download_events` *Download Event Data*

Description

Downloads patient event data. Typically done after all records have been annotated and the project is complete.

Usage

```
download_events(uri_fun, user, password, host, port, database)
```

Arguments

`uri_fun`	Uniform resource identifier (URI) string generating function for MongoDB credentials.
`user`	MongoDB user name.
`password`	MongoDB user password.
`host`	MongoDB host server.
`port`	MongoDB port.
`database`	MongoDB database name.

Examples

```
## Not run:
download_events(uri_fun = mongo_uri_standard, user = 'John', password = 'db_password_1234',
host = 'server1234', port = NA)

## End(Not run)
```

end_users *Download End User List*

Description

Downloads list of CEDARS end users along with their passwords.

Usage

```
end_users(uri_fun, user, password, host, port, database)
```

Arguments

`uri_fun`	Uniform resource identifier (URI) string generating function for MongoDB credentials.
`user`	MongoDB user name.
`password`	MongoDB user password.
`host`	MongoDB host server.
`port`	MongoDB port.
`database`	MongoDB database name.

Examples

```
## Not run:
end_users(uri_fun = mongo_uri_standard, user = 'John', password = 'db_password_1234',
host = 'server1234', port = NA, database = 'TEST_PROJECT')

## End(Not run)
```

find_project_name *Generate unique test project name (i.e. DB name) on MongoDB CEDARS testing cluster*

Description

Parses existing DB names and randomly generates a unique test project name on MongoDB CEDARS testing cluster. This is used for convenience purposes when the R user does not have an existing MongoDB connection. The corresponding database and collections are PUBLIC so no patient information or any other privileged/confidential data should be used! This is for testing on simulated records only.

Usage

```
find_project_name()
```

Examples

```
## Not run:
find_project_name()

## End(Not run)
```

get_model *Get a NLP Model*

Description

Downloads a NLP model, presently only UDPipe models supported.

Usage

```
get_model(model_name = "english-ewt", platform = "udpipe")
```

Arguments

`model_name`	Name of models to download.
`platform`	Name of NLP platform, currently only UDPipe is supported.

Value

Saves model in inst/models.

get_wrapper *Wrap the get_data() Function*

Description

Obtain one sentence and related info from MongoDB. Uses DB credentials pre-loaded in the main environment. For use with Shiny or REST GET (latter yet to be implemented).

Usage

```
get_wrapper(
  database,
  end_user,
  end_user_password,
  html = TRUE,
  position,
  patient_id = NA,
  ldap = FALSE
)
```

Arguments

`database`	MongoDB database.
`end_user`	CEDARS end user name..
`end_user_password`	CEDARS end user password.
`html`	Should output keywords/concepts be highlighted with HTML markup? Default is TRUE.

`position`	Sentence position within the sequence of selected sentences for a given patient.
`patient_id`	Used if a specific patient record is requested, instead of a search for next record to annotate.
`ldap`	Is LDAP authentication being used? If so, password will not be checked and access will be granted automatically.

Value

A list with patient-specific information and a dataframe with selected sentences along with sentence-specific data.

Examples

```
## Not run:
get_wrapper(database = 'TEST_PROJECT', end_user = 'John', end_user_password = 'db_password_1234',
html = TRUE, position = NA)

## End(Not run)
```

`initialize_annotations` *Initialize Annotations Deletes all NLP annotations and patient-specific information, including clinical event dates.New, empty 'ANNOTATIONS' and 'PATIENTS' collections are created. Dictionaries and original patient notes are preserved.*

Description

Initialize Annotations Deletes all NLP annotations and patient-specific information, including clinical event dates.New, empty 'ANNOTATIONS' and 'PATIENTS' collections are created. Dictionaries and original patient notes are preserved.

Usage

```
initialize_annotations(uri_fun, user, password, host, port, database)
```

Arguments

`uri_fun`	Uniform resource identifier (URI) string generating function for MongoDB credentials.
`user`	MongoDB user name.
`password`	MongoDB user password.
`host`	MongoDB host server.
`port`	MongoDB port.
`database`	MongoDB database name.

Examples

```
## Not run:
initialize_annotations(uri_fun = mongo_uri_standard, user = 'John', password = 'db_password_1234',
host = 'server1234', port = NA, database = 'TEST_PROJECT')

## End(Not run)
```

`initialize_notes` *Initialize EHR Notes*

Description

Deletes all patient notes from the database.

Usage

```
initialize_notes(uri_fun, user, password, host, port, database)
```

Arguments

`uri_fun`	Uniform resource identifier (URI) string generating function for MongoDB credentials.
`user`	MongoDB user name.
`password`	MongoDB user password.
`host`	MongoDB host server.
`port`	MongoDB port.
`database`	MongoDB database name.

Examples

```
## Not run:
initialize_notes(uri_fun = mongo_uri_standard, user = 'John', password = 'db_password_1234',
host = 'server1234', port = NA, database = 'TEST_PROJECT')

## End(Not run)
```

`initialize_patients` *Initialize Patient List*

Description

All patient-specific information is deleted, including clinical event dates. Original notes and NLP annotations are preserved.

Usage

```
initialize_patients(uri_fun, user, password, host, port, database)
```

Arguments

`uri_fun`	Uniform resource identifier (URI) string generating function for MongoDB credentials.
`user`	MongoDB user name.
`password`	MongoDB user password.
`host`	MongoDB host server.
`port`	MongoDB port.
`database`	MongoDB database name.

Examples

```
## Not run:
initialize_patients(uri_fun = mongo_uri_standard, user = 'John', password = 'db_password_1234',
host = 'server1234', port = NA, database = 'TEST_PROJECT')

## End(Not run)
```

`initialize_users` *Initialize End User List*

Description

Deletes all CEDARS end user credentials information.

Usage

```
initialize_users(uri_fun, user, password, host, port, database)
```

Arguments

`uri_fun`	Uniform resource identifier (URI) string generating function for MongoDB credentials.
`user`	MongoDB user name.
`password`	MongoDB user password.
`host`	MongoDB host server.
`port`	MongoDB port.
`database`	MongoDB database name.

Examples

```
## Not run:
initialize_users(uri_fun = mongo_uri_standard, user = 'John', password = 'db_password_1234',
host = 'server1234', port = NA, database = 'TEST_PROJECT')

## End(Not run)
```

`mongo_uri_standard` *Prepare MongoDB URI string, most commonly used format*

Description

Formats the MongoDB URI string for use by package mongolite. In this case the 'standard' URI format is used.

Usage

```
mongo_uri_standard(user, password, host, port = NA)
```

Arguments

`user`	MongoDB user name.
`password`	MongoDB user password.
`host`	MongoDB host server.
`port`	MongoDB port.

Value

URI string.

Examples

```
## Not run:
mongo_uri_standard(user = 'John', password = 'db_password_1234', host = 'server1234', port = NA)

## End(Not run)
```

`mrconso_upload` *Upload UMLS Dictionary*

Description

Prepares and uploads UMLS MRCONSO.RRF file. This file is not included in the CEDARS package and can be obtained on the NIH web site at https://www.nlm.nih.gov/research/umls/index.html.

Usage

```
mrconso_upload(
  path,
  language = "ENG",
  subsets,
  max_grams = 7,
  uri_fun,
  user,
  password,
  host,
  port,
  database
)
```

Arguments

`path`	Path to file MRCONSO.RRF.
`language`	Language of biomedical lexicon, default is English (ENG).
`subsets`	Character vector of lexicon subsets to retain. UMLS is quite large so most applications can use only a few lexicon subsets.
`max_grams`	Maximum length of token in grams. Tokens above the thresold length will not be retained. Empirically, a value of 7 suffices for most applications.
`uri_fun`	Uniform resource identifier (URI) string generating function for MongoDB credentials.
`user`	MongoDB user name.
`password`	MongoDB user password.
`host`	MongoDB host server.
`port`	MongoDB port.
`database`	MongoDB database name.

Examples

```
## Not run:
mrconso_upload(path = 'dictionaries/MRCONSO.RRF', language = 'ENG', subsets = c('SNOMEDCT_US',
'MTHICD9', 'ICD9CM', 'ICD10', 'ICD10CM', 'DSM-5', 'MSH', 'RXNORM', 'NCI'), max_grams = 7,
user = 'John', password = 'db_password_1234', host = 'server1234', port = NA,
database = 'TEST_PROJECT')

## End(Not run)
```

`mrrel_upload` *Upload UMLS Relationships*

Description

Prepares and uploads UMLS MRREL.RRF file. This file is not included in the CEDARS package and can be obtained on the NIH web site at https://www.nlm.nih.gov/research/umls/index.html. It is very large and not currently used by CEDARS.

Usage

```
mrrel_upload(path, uri_fun, user, password, host, port, database)
```

Arguments

`path`	Path to file MRREL.RRF.
`uri_fun`	Uniform resource identifier (URI) string generating function for MongoDB credentials.
`user`	MongoDB user name.
`password`	MongoDB user password.
`host`	MongoDB host server.
`port`	MongoDB port.
`database`	MongoDB database name.

Examples

```
## Not run:
mrrel_upload(path = 'dictionaries/MRREL.RRF', uri_fun = mongo_uri_standard, user = 'John',
password = 'db_password_1234', host = 'server1234', port = NA, database = 'TEST_PROJECT')

## End(Not run)
```

`negex` *Negex data Apache License 2.0*

Description

Negex data Apache License 2.0

Usage

```
data(negex)
```

Format

An object of class `'data.frame'`.

Source

Google Code Archive

References

Chapman et al. (2013) Stud Health Technol Inform 192:677-681 (PubMed)

Examples

```
data(negex)
```

`negex_upload` *Upload NegEx*

Description

Prepares and uploads NegEx negation lexicon. It is not absolutely required for CEDARS to function but in practice will improve search accuracy for most applications.

Usage

```
negex_upload(
  uri_fun,
  user,
  password,
  host,
  port,
  database,
  selected_model_path = NA
)
```

Arguments

`uri_fun`	Uniform resource identifier (URI) string generating function for MongoDB credentials.
`user`	MongoDB user name.
`password`	MongoDB user password.
`host`	MongoDB host server.
`port`	MongoDB port.
`database`	MongoDB database name.
`selected_model_path`	Path to NLP model file.

Examples

```
## Not run:
negex_upload(uri_fun = mongo_uri_standard, user = 'John', password = 'db_password_1234',
host = 'server1234', port = NA, database = 'TEST_PROJECT', NA)

## End(Not run)
```

`post_wrapper` *Wrap the post_data() Function*

Description

Posts results of human reviewer annotation to MongoDB. Uses DB credentials pre-loaded in the main environment. For use with Shiny or REST POST (latter yet to be implemented).

Usage

```
post_wrapper(
  database,
  end_user,
  end_user_password,
  position,
  event_date,
  pt_comments,
  ldap = FALSE
)
```

Arguments

`database`	MongoDB database.
`end_user`	CEDARS end user name.
`end_user_password`	CEDARS end user password.
`position`	Sentence position within the sequence of selected sentences for a given patient.
`event_date`	Date of clinical event as determined by human reviewer.
`pt_comments`	Patient-specific comments from the reviewer.
`ldap`	Is LDAP authentication being used? If so, password will not be checked and access will be granted automatically.

Examples

```
## Not run:
post_wrapper(database = 'TEST_PROJECT', end_user = 'John', end_user_password = 'db_password_1234',
position = NA, event_date = NA, pt_comments = 'This is a comment')

## End(Not run)
```

`pre_search` *Execute Search on a Set of Records*

Description

Batches a keyword/CUI search for a cohort of patients. Useful to speed up the process by end users, since search results will be pre-populated. Locks each record before proceeding with search on existing NLP annotations. Patient records with no matching sentences or a known event date at or before the earliest matching sentence will be marked as reviewed. The latter assumes the query orders to skip sentences after events.

Usage

```
pre_search(patient_vect = NA, uri_fun, user, password, host, port, database)
```

Arguments

`patient_vect`	Vector of patient ID's. Default is NA, in which case all unlocked records will be searched.
`uri_fun`	Uniform resource identifier (URI) string generating function for MongoDB credentials.
`user`	MongoDB user name.
`password`	MongoDB user password.
`host`	MongoDB host server.
`port`	MongoDB port.
`database`	MongoDB database name.

Examples

```
## Not run:
pre_search(patient_vect = NA, uri_fun = mongo_uri_standard, user = 'John',
password = 'db_password_1234', host = 'server1234', database = 'TEST_PROJECT')

## End(Not run)
```

---

save_credentials *Save MongoDB Credentials*

Description

Saves MongoDB credentials as 'db_credentials.Rdata' and Shiny app file as 'app.R'. Those two files should be copied to the Shiny Server app directory. Needed only if using Shiny Server; credentials are entered in the command line for local app use.

Usage

```
save_credentials(
  user,
  password,
  host,
  port,
  database,
  LDAP,
  destination_path = getwd()
)
```

Arguments

user	MongoDB user name.
password	MongoDB user password.
host	MongoDB server host.
port	MongoDB port.
database	MongoDB database name.
LDAP	is LDAP being used? In this case, CEDARS will not prompt for user ID/password and a check will NOT be made on the users table. Access will be granted, relying on LDAP authentication. Annotations will be stamped with LDAP user name.
destination_path	Folder where the files should be saved. Default is working directory.

Examples

```
## Not run:
save_credentials(user = 'John', password = 'db_password_1234', host = 'server1234',
database = 'myDB', LDAP = FALSE, destination_path = getwd())

## End(Not run)
```

`save_query` *Save Search Query*

Description

Saves the search query. The query consists of keywords/UMLS concept unique identifiers (CUI's), boolean elements and other operators ('AND', 'OR', '!', '(', or ')').

Usage

```
save_query(
  uri_fun,
  user,
  password,
  host,
  port,
  database,
  search_query,
  use_negation,
  hide_duplicates,
  skip_after_event
)
```

Arguments

`uri_fun`	Uniform resource identifier (URI) string generating function for MongoDB credentials.
`user`	MongoDB user name.
`password`	MongoDB user password.
`host`	MongoDB host server.
`port`	MongoDB port.
`database`	MongoDB database name.
`search_query`	Medical corpus query containg keywords/CUI's, boolean elements and other operators ('AND', 'OR', '!', '(', or ')').
`use_negation`	Should negated items be ignored in the keyword/concept search?
`hide_duplicates`	Should duplicated sentences be removed for search results?
`skip_after_event`	Should sentences occurring after recorded clinical event be skipped?

Examples

```
## Not run:
save_query(uri_fun = mongo_uri_standard, user = 'John', password = 'db_password_1234',
host = 'server1234', port = NA, database = 'TEST_PROJECT', search_query = 'thrombosis AND venous',
use_negation = TRUE, hide_duplicates = TRUE, skip_after_event = TRUE)

## End(Not run)
```

save_tags *Save Document Tags*

Description

Save name of EHR document metadata tags. Individual notes or parts of notes can be labelled with up to 10 tags, typically the patient's name at the time, the type of note, the note section, the author, etc. Tags are not mandatory.

Usage

```
save_tags(uri_fun, user, password, host, port, database, tag_vect)
```

Arguments

`uri_fun`	Uniform resource identifier (URI) string generating function for MongoDB credentials.
`user`	MongoDB user name.
`password`	MongoDB user password.
`host`	MongoDB host server.
`port`	MongoDB port.
`database`	MongoDB database name.
`tag_vect`	Character vector of 10 tag names.

Examples

```
## Not run:
save_tags(uri_fun = mongo_uri_standard, user = 'John', password = 'db_password_1234',
host = 'server1234', port = NA, database = 'TEST_PROJECT',
tag_vect = c('note_type', 'note_section', 'author', 'patient_name', NA, NA, NA, NA, NA, NA))

## End(Not run)
```

simulated_patients *Simulated patient data GPL-3 license*

Description

Simulated patient data GPL-3 license

Usage

```
data(simulated_patients)
```

Format

An object of class `'data.frame'`.

Examples

```
data(simulated_patients)
```

`start_local` *Start CEDARS Locally*

Description

Starts CEDARS locally from RStudio. This is a functional approach and is easier to implement than a full-fledged Shiny Server. Multiple users can access the same CEDARS project on the MongoDB server using separate local R sessions, however in that case MongoDB credentials would have to be shared to all. The best option for multi-user implementations is to use Shiny Server.

Usage

```
start_local(user, password, host, port, database)
```

Arguments

`user`	DB user name.
`password`	DB password.
`host`	Host server.
`port`	MongoDB port.
`database`	MongoDB database name.

Examples

```
## Not run:
start_local(user = 'John', password = 'db_password_1234', host = 'server1234', port = NA,
database = 'myDB')

## End(Not run)
```

`terminate_project` *Terminate CEDARS Project*

Description

Everything is removed, including dictionaries. MongoDB account used must have sufficient privileges.

Usage

```
terminate_project(uri_fun, user, password, host, port, database, fast = FALSE)
```

Arguments

`uri_fun`	Uniform resource identifier (URI) string generating function for MongoDB credentials.
`user`	MongoDB user name.
`password`	MongoDB user password.
`host`	MongoDB host server.
`port`	MongoDB port.
`database`	MongoDB database name.
`fast`	If TRUE, delete everything without asking security questions.

Examples

```
## Not run:
terminate_project(uri_fun = mongo_uri_standard, user = 'John', password = 'db_password_1234',
host = 'server1234', port = NA, database = 'TEST_PROJECT')

## End(Not run)
```

`unlock_user` *Unlock User-Specific Records Removes any pending lock(s) for a specific user. Normally there should not be more than one record locked per user at any given time, but if there were more than one, i.e. DB corruption, all locks would be lifted at once.*

Description

Unlock User-Specific Records Removes any pending lock(s) for a specific user. Normally there should not be more than one record locked per user at any given time, but if there were more than one, i.e. DB corruption, all locks would be lifted at once.

Usage

```
unlock_user(uri_fun, user, password, host, port, database, end_user)
```

Arguments

`uri_fun`	Uniform resource identifier (URI) string generating function for MongoDB credentials.
`user`	MongoDB user name.
`password`	MongoDB user password.
`host`	MongoDB host server.
`port`	MongoDB port.
`database`	MongoDB database name.
`end_user`	CEDARS end user.

Examples

```
## Not run:
unlock_user(uri_fun = mongo_uri_standard, user = 'John', password = 'db_password_1234',
host = 'server1234', port= NA, database = 'TEST_PROJECT', end_user = 'Mike')

## End(Not run)
```

`upload_events` *Upload Event Data*

Description

Uploads event dates for patients already in the patient list. Useful when some events have already been documented before runnning CEDARS, for example as a second-line method to catch events missed with a different approach. Only event dates for existing records are altered, missing patient records are not added!

Usage

```
upload_events(
  uri_fun,
  user,
  password,
  host,
  port,
  database,
  patient_ids,
  event_dates
)
```

Arguments

`uri_fun`	Uniform resource identifier (URI) string generating function for MongoDB credentials.
`user`	MongoDB user name.
`password`	MongoDB user password.
`host`	MongoDB host server.
`port`	MongoDB port.
`database`	MongoDB database name.
`patient_ids`	Vector of patient ID’s.
`event_dates`	Vector of clinical event dates.

Examples

```
## Not run:
upload_events(uri_fun = mongo_uri_standard, user = 'John', password = 'db_password_1234',
host = 'server1234', port = NA, database = 'TEST_PROJECT', patient_ids = ids, event_dates = events)

## End(Not run)
```

upload_notes *Upload Notes to Database*

Description

Allows user to populate notes in database from dataframe; could be easily inserted into wrapper batch function to serially download from other DB etc. Notes dataframe must contain: 'patient_id', 'text_id' (a unique identifier for each text segment), along with 'text', 'text_date', 'doc_id' (designates unique EHR document) and ideally 'text_sequence' which indicates order of text section within document. 'doc_section_name' along 'text_tag_1' to 'text_tag_10' are optional. 'text_date' must be in format '%Y-%m-%d'!

Usage

```
upload_notes(uri_fun, user, password, host, port, database, notes)
```

Arguments

`uri_fun`	Uniform resource identifier (URI) string generating function for MongoDB credentials.
`user`	MongoDB user name.
`password`	MongoDB user password.
`host`	MongoDB host server.
`port`	MongoDB port.
`database`	MongoDB database name.
`notes`	Dataframe of EHR documents with metadata. The documents can consist of full notes or note subsections.

Examples

```
## Not run:
upload_notes(uri_fun = mongo_uri_standard, user = 'John', password = 'db_password_1234',
host = 'server1234', port = NA, database = 'TEST_PROJECT', notes = simulated_patients)

## End(Not run)
```

Index