Package 'medExtractR'

July 22, 2025

Version 0.4.1

Date 2022-06-06

Title Extraction of Medication Information from Clinical Text

Description Function and support for medication and dosing information extraction from free-text clinical notes. Medication entities for the basic medExtractR implementation that can be extracted include drug name, strength, dose amount, dose, frequency, intake time, dose change, and time of last dose. The basic medExtractR is outlined in Weeks, Beck, McNeer, Williams, Bejan, Denny, Choi (2020) <doi:10.1093/jamia/ocz207>. The extended medExtractR_tapering implementation is intended to extract dosing information for more tapering schedules, which are far more complex. The tapering extension allows for the extraction of additional entities including dispense amount, refills, dose schedule, time keyword, transition, and preposition.

License GPL (>= 2)

Depends R (>= 2.10)

Encoding UTF-8

LazyData true

Imports stringi, stringr

Suggests knitr, ggplot2, rmarkdown, markdown, parallel

RoxygenNote 7.1.1

VignetteBuilder knitr

NeedsCompilation no

Author Leena Choi [aut, cre] (ORCID: <https://orcid.org/0000-0002-2544-7090>), Cole Beck [aut] (ORCID: <https://orcid.org/0000-0002-6849-6255>), Hannah Weeks [aut] (ORCID: <https://orcid.org/0000-0002-0262-6790>)

Maintainer Leena Choi <leena.choi@vanderbilt.edu>

Repository CRAN

Date/Publication 2022-06-06 22:40:02 UTC

Contents

medExtractR-package . . . 2
addl_expr . . . 3
dosechange_vals . . . 3
doseschedule_vals . . . 4
duration_vals . . . 4
extract_entities . . . 5
extract_entities_tapering . . . 7
extract_generic . . . 10
extract_lastdose . . . 11
frequency_vals . . . 12
intaketime_vals . . . 13
medExtractR . . . 13
medExtractR_tapering . . . 16
preposition_vals . . . 18
route_vals . . . 19
rxnorm_druglist . . . 19
string_counts . . . 20
string_occurs . . . 21
string_suggestions . . . 22
timekeyword_vals . . . 22
time_regex . . . 23
transition_vals . . . 24

Index **25**

`medExtractR-package` *Medication Extraction With R*

Description

Provides a function `medExtractR` for extracting dose attributes for medications within a given electronic health record (EHR) note.

Author(s)

Hannah Weeks <hannah.l.weeks@vanderbilt.edu>,
Cole Beck <cole.beck@vumc.org>,
Leena Choi <leena.choi@vumc.org>

Maintainer: Leena Choi <leena.choi@vumc.org>

Examples

```
note1 <- "Progrf Oral Capsule 1 mg 3 capsules by mouth twice a day - last
dose at 10pm"
note2 <- "Currently on lamotrigine 150-200, but will increase to lamotrigine 200mg bid"
medExtractR(note1, c("prograf", "tacrolimus"), 60, "mg", 2, lastdose=TRUE)
medExtractR(note2, c("lamotrigine", "ltg"), 130, "mg", 1, strength_sep = "-")
```

---

addl_expr *Additional expressions for* drug_list

Description

A dictionary with additional expressions that can be used to supplement the drug_list argument of medExtractR and medExtractR_tapering.

Usage

```
addl_expr
```

Format

A data frame with the following variables:

expr A character vector, additional optional expressions for the drug_list argument.

type A character vector, what category the expression belongs to (e.g., symptom, lab name, medication abbreviation, or drug class).

Examples

```
data(addl_expr)
```

---

dosechange_vals *Keywords Specifying Dose Change*

Description

A dictionary of words indicating a dose change, meaning that the associated drug regimen may not be current. This includes phrases such as increase, reduce, or switch. In the following example of clinical text, the word 'increase' represents a dose change keyword: "Increase prograf to 5mg bid."

Usage

```
dosechange_vals
```

Format

A data frame with dose change expressions (exact and/or regular expressions).

expr A character vector, expressions to consider as dose change.

Examples

```
data(dosechange_vals)
```

---

doseschedule_vals *Keywords Specifying Dose Schedule*

Description

A dictionary with words for indicating a tapering dosing schedule. These can explicitly refer to such a schedule with phrases like "tapering" or "wean". It also includes words indicating an alternating dose schedule (e.g., "alternate", "alt.", "even days", or "odd days") as well as stopping keywords indicating the patient is going completely off the medication (e.g., "done", "gone", "stop", "discontinue").

Usage

```
doseschedule_vals
```

Format

A data frame with dose schedule expressions (exact and/or regular expressions).

expr A character vector, expressions to consider as dose schedule.

Examples

```
data(doseschedule_vals)
```

---

duration_vals *Keywords Specifying Duration*

Description

A dictionary with phrases indicating how long the patient should take a particular dose of the drug. Examples of duration expressions include "2 weeks", "14 days", "another 3 days", "through mid-April", or a specific date. The form of each duration is given as a regular expression.

Usage

```
duration_vals
```

Format

A data frame with duration expressions (exact and/or regular expressions).

expr A character vector, expressions to consider as duration.

Examples

```
data(duration_vals)
```

`extract_entities` *Extract Medication Entities From Phrase*

Description

This function searches a phrase for medication dosing entities of interest. It is called within `medExtractR` and generally not intended for use outside that function. The `phrase` argument containing text to search corresponds to an individual mention of the drug of interest.

Usage

```
extract_entities(
  phrase,
  p_start,
  p_stop,
  unit,
  frequency_fun = NULL,
  intaketime_fun = NULL,
  duration_fun = NULL,
  route_fun = NULL,
  strength_sep = NULL,
  ...
)
```

Arguments

`phrase`	Text to search.
`p_start`	Start position of phrase within original text.
`p_stop`	End position of phrase within original text.
`unit`	Unit of measurement for medication strength, e.g. 'mg'.
`frequency_fun`	Function used to extract frequency.
`intaketime_fun`	Function used to extract intake time.
`duration_fun`	Function used to extract duration.
`route_fun`	Function used to extract route.
`strength_sep`	Delimiter for contiguous medication strengths.

... Parameter settings used in extracting frequency and intake time, including additional arguments to the <entity>_fun arguments. Use frequency_dict, intaketime_dict, duration_dict, and route_dict to identify custom frequency, intake time, duration, and route dictionaries, respectively.

Details

Various medication dosing entities are extracted within this function including the following:

strength: The amount of drug in a given dosage form (i.e., tablet, capsule).
dose amount: The number of tablets, capsules, etc. taken at a given intake time.
dose strength: The total amount of drug given intake. This quantity would be equivalent to strength x dose amount, and appears similar to strength when dose amount is absent.
frequency: The number of times per day a dose is taken, e.g., "once daily" or '2x/day'.
intaketime: The time period of the day during which a dose is taken, e.g., 'morning', 'lunch', 'in the pm'.
duration: How long a patient is on a drug regimen, e.g., '2 weeks', 'mid-April', 'another 3 days'.
route: The administration route of the drug, e.g., 'by mouth', 'IV', 'topical'.

Note that extraction of the entities drug name, dose change, and time of last dose are not handled by the extract_entities function. Those entities are extracted separately and appended to the extract_entities output within the main medExtractR function. Strength, dose amount, and dose strength are primarily numeric quantities, and are identified using a combination of regular expressions and rule-based approaches. Frequency, intake time, route, and duration, on the other hand, use dictionaries for identification.

By default and when an argument <entity>_fun is NULL, the extract_generic function will be used to extract that entity. This function can also inherit user-defined entity dictionaries, supplied as arguments <entity>_dict to medExtractR or medExtractR_tapering (see documentation files for main function(s) for details).

The stength_sep argument is NULL by default, but can be used to identify shorthand for morning and evening doses. For example, consider the phrase "Lamotrigine 300-200" (meaning 300 mg in the morning and 200 mg in the evening). The argument strength_sep = '-' identifies the full expression *300-200* as *dose strength* in this phrase.

Value

data.frame with entities information. At least one row per entity is returned, using NA when no expression was found for a given entity.
The "entity" column of the output contains the formatted label for that entity, according to the following mapping.
strength: "Strength"
dose amount: "DoseAmt"
dose strength: "DoseStrength"
frequency: "Frequency"
intake time: "IntakeTime"
duration: "Duration"
route: "Route"
Sample output for the phrase "Lamotrigine 200mg bid" would look like:

entity	expr
IntakeTime	<NA>
Strength	<NA>
DoseAmt	<NA>
Route	<NA>
Duration	<NA>
Frequency	bid;19:22
DoseStrength	200mg;13:18

Examples

```
note <- "Lamotrigine 25 mg tablet - 3 tablets oral twice daily"
extract_entities(note, 1, nchar(note), "mg")
# A user-defined dictionary can be used instead of the default
my_dictionary <- data.frame(c("daily", "twice daily"))
extract_entities(note, 1, 53, "mg", frequency_dict = my_dictionary)
```

`extract_entities_tapering` *Extract Medication Entities From Phrase - Extension of* `extract_entities` *for Tapering application*

Description

This function searches a phrase for medication dosing entities of interest. It is called within `medExtractR_tapering` and generally not intended for use outside that function.

Usage

```
extract_entities_tapering(
  phrase,
  p_start,
  d_stop,
  unit,
  frequency_fun = NULL,
  intaketime_fun = NULL,
  duration_fun = NULL,
  route_fun = NULL,
  doseschedule_fun = NULL,
  preposition_fun = NULL,
  timekeyword_fun = NULL,
  transition_fun = NULL,
  dosechange_fun = NULL,
  strength_sep = NULL,
  ...
)
```

Arguments

phrase	Text to search.
p_start	Start position of phrase within original text.
d_stop	End position of drug name within original text.
unit	Unit of measurement for medication strength, e.g., 'mg'.
frequency_fun	Function used to extract frequency.
intaketime_fun	Function used to extract intake time.
duration_fun	Function used to extract duration.
route_fun	Function used to extract route.
doseschedule_fun	Function used to extract dose schedule.
preposition_fun	Function used to extract preposition.
timekeyword_fun	Function used to extract time keyword.
transition_fun	Function used to extract transition.
dosechange_fun	Function used to extract dose change.
strength_sep	Delimiter for contiguous medication strengths.
...	Parameter settings used in extracting frequency and intake time, including additional arguments to frequency_fun and intaketime_fun. Use frequency_dict to identify custom frequency dictionaries and intaketime_dict to identify custom intake time dictionaries. Similarly, for all other entities with a corresponding <entity>_fun, a custom dictionary can be supplied with the argument <entity>_dict.

Details

Various medication dosing entities are extracted within this function including the following:

strength: The amount of drug in a given dosage form (i.e., tablet, capsule).
dose amount: The number of tablets, capsules, etc. taken at a given intake time.
dose strength: The total amount of drug given intake. This quantity would be equivalent to strength x dose amount, and appears similar to strength when dose amount is absent.
frequency: The number of times per day a dose is taken, e.g., "once daily" or '2x/day'.
intaketime: The time period of the day during which a dose is taken, e.g., 'morning', 'lunch', 'in the pm'.
duration: How long a patient is on a drug regimen, e.g., '2 weeks', 'mid-April', 'another 3 days'.
route: The administration route of the drug, e.g., 'by mouth', 'IV', 'topical'.
dose change: Whether the dosage of the drug was changed, e.g., 'increase', 'adjust', 'reduce'.
dose schedule: Keywords which represent special dosing regimens, such as tapering schedules, alternating doses, or stopping keywords, e.g., 'weaning', 'even days' or 'odd_days', 'discontinue'.
time keyword: Whether the dosing regimen is a past dose, current dose, or future dose, e.g., 'currently', 'remain', 'yesterday'.
transition: Words or symbols that link consecutive doses of a tapering regimen, e.g., 'then', 'followed by', or a comma ','.

preposition: Prepositions that occur immediately next to another identified entity, e.g., 'to', 'until', 'for'.
dispense amount: The number of pills prescribed to the patient.
refill: The number of refills allowed for the patient's prescription.

Similar to the basic implementation, drug name and and time of last dose are not handled by the `extract_entities_tapering` function. Those entities are extracted separately and appended to the `extract_entities_tapering` output within the main `medExtractR_tapering` function. In the tapering extension, however, dose change is treated the same as other dictionary-based entities and extracted within `extract_entities_tapering`. Strength, dose amount, dose strength, dispense amount, and refill are primarily numeric quantities, and are identified using a combination of regular expressions and rule-based approaches. All other entities use dictionaries for identification. For more information about the default dictionary for a specific entity, view the documentation file for the object `<entity>_vals`.

By default and when an argument `<entity>_fun` is `NULL`, the `extract_generic` function will be used to extract that entity. This function can also inherit user-defined entity dictionaries for each entity, supplied as arguments `<entity>_dict` to `medExtractR` or `medExtractR_tapering` (see documentation files for main function(s) for details).

Note that `extract_entities_tapering` has the argument `d_stop`. This differs from `extract_entities`, which uses the end position of the full search window. This is a consequence of `medExtractR` using a fixed search window length and `medExtractR_tapering` dynamically constructing a search window.

Value

data.frame with entities information. At least one row per entity is returned, using `NA` when no expression was found for a given entity.
The "entity" column of the output contains the formatted label for that entity, according to the following mapping.
strength: "Strength"
dose amount: "DoseAmt"
dose strength: "DoseStrength"
frequency: "Frequency"
intake time: "IntakeTime"
duration: "Duration"
route: "Route"
dose change: "DoseChange"
dose schedule: "DoseScheule"
time keyword: "TimeKeyword"
transition: "Transition"
preposition: "Preposition"
dispense amount: "DispenseAmt"
refill: "Refill"
Sample output for the phrase "Lamotrigine 200mg bid for 14 days" would look like:

entity	expr
IntakeTime	<NA>

Strength	<NA>
DoseAmt	<NA>
DoseChange	<NA>
DoseSchedule	<NA>
TimeKeyword	<NA>
Transition	<NA>
Preposition	<NA>
DispenseAmt	<NA>
Refill	<NA>
Frequency	bid;19:22
DoseStrength	200mg;13:18
Preposition	for;23:26
Duration	14 days;27:34

Examples

```
note <- "prednisone 20mg daily tapering to 5mg daily over 2 weeks"
extract_entities_tapering(note, 1, 11, "mg")
# A user-defined dictionary can be used instead of the default
my_dictionary <- data.frame(c("daily", "twice daily"))
extract_entities(note, 1, 11, "mg", frequency_dict = my_dictionary)
```

---

extract_generic *Extract Generic Entities From Phrase*

Description

This function searches a phrase for the position and length of expressions specified in a dictionary. This is called within other main functions of the package and generally not intended for use on its own.

Usage

```
extract_generic(phrase, dict)
```

Arguments

phrase	Text to search.
dict	data.frame, the first column should contain expressions to find. These can be regular expressions or exact phrases.

Details

extract_generic is used to extract entities that are identified with an associated dictionary of phrases or regular expressions, such as dose change, frequency, intake time, route, or duration in medExtractR and medExtractR_tapering, as well as dose schedule, time keyword, transition, and preposition in medExtractR_tapering. This function is called within extract_entities.

Value

A numeric matrix with position and expression length.

Examples

```
data(frequency_vals)
extract_generic("take two every day", dict = frequency_vals)
extract_generic("take two every morning",
                  dict = data.frame(c("morning", "every morning")))
```

`extract_lastdose` *Extract Last Dose Time From Phrase*

Description

This function searches a phrase for the expression and postion of the time at which the last dose of a drug was taken. It is called within `medExtractR` and generally not intended for use outside that function.

Usage

```
extract_lastdose(phrase, p_start, d_start, d_stop, time_exp = "default")
```

Arguments

`phrase`	Text to search.
`p_start`	Start position of phrase in the overall text (e.g., the full clinical note).
`d_start`	Start position of drug name in larger text.
`d_stop`	End position of drug name in larger text.
`time_exp`	Vector of regular expressions to identify time expressions.

Details

This function identifies the time at which the last dose of a drug of interest was taken. The arguments `p_start`, `d_start`, and `d_stop` represent global start or stop positions for the phrase or drug. These arguments are used to determine the position of any found last dose time expressions relative to the overall clinical note, not just within `phrase`.

The `time_exp` argument contains regular expressions for numeric or text representations of last dose time. See `time_regex` for more information about the default regular expressions used in `medExtractR`.

Value

data.frame with last dose time entity information. This output format is consistent with the output of `extract_entities`, and the formatted label for the time of last dose entity is "LastDose." Sample output for the phrase “Last prograf at 5pm” would look like:

entity	expr
LastDose	5pm;17:20

Examples

```
# Suppose this phrase begins at character 120 in the overall clinical note
extract_lastdose("took aspirin last night at 8pm", p_start = 120,
                  d_start = 125, d_stop = 131)
```

`frequency_vals` *Keywords Specifying Frequency*

Description

A dictionary mapping frequency expressions to numeric values representing the corresponding number of doses per day. Example expressions include "q12 hours", "bid", "daily", and "three times a day". The form of each frequency is given as a regular expression.

Usage

```
frequency_vals
```

Format

A data frame with frequency expressions (exact and/or regular expressions).

expr A character vector, expressions to consider as frequency.

value A numeric vector, numeric value of frequency represented as number of doses taken per day. For example, “bid” and “twice a day” would both have a numeric value of 2.

Examples

```
data(frequency_vals)
```

intaketime_vals *Keywords Specifying Intake Time*

Description

A dictionary with intake time expressions representing the approximate time of day when a dose should be taken. Example expressions include "in the morning", "with lunch", "at bedtime", and "qpm". The form of each intake time is given as a regular expression.

Usage

```
intaketime_vals
```

Format

A data frame with intake time expressions (exact and/or regular expressions).

expr A character vector, expressions to consider as intake time.

Examples

```
data(intaketime_vals)
```

medExtractR *Extract Medication Entities From Clinical Note*

Description

This function identifies medication entities of interest and returns found expressions with start and stop positions.

Usage

```
medExtractR(
  note,
  drug_names,
  window_length,
  unit,
  max_dist = 0,
  drug_list = "rxnorm",
  lastdose = FALSE,
  lastdose_window_ext = 1.5,
  strength_sep = NULL,
  flag_window = 30,
  dosechange_dict = "default",
  ...
)
```

Arguments

note	Text to search.
drug_names	Vector of drug names of interest to locate.
window_length	Length (in number of characters) of window after drug in which to look.
unit	Strength unit to look for (e.g., 'mg').
max_dist	Numeric - edit distance to use when searching for drug_names.
drug_list	Vector of known drugs that may end search window. By default calls rxnorm_druglist.
lastdose	Logical - whether or not last dose time entity should be extracted.
lastdose_window_ext	Numeric - multiplicative factor by which window_length should be extended when identifying last dose time.
strength_sep	Delimiter for contiguous medication strengths (e.g., '-' for "LTG 200-300").
flag_window	How far around drug (in number of characters) to look for dose change keyword - default fixed to 30. See 'Details' section below for further explanation.
dosechange_dict	List of keywords used to determine if a dose change entity is present.
...	Parameter settings used in extracting frequency, intake time, route, and duration. Potentially useful parameters include freq_dict, intaketime_dict, route_dict, and duration_dict (see ... argument in extract_entities) to specify frequency or intake time dictionaries, as well as 'freq_fun', 'intaketime_fun', 'route_fun', and 'duration_fun' for user-specified extraction functions. If no additional arguments are provided, medExtractR_tapering will use extract_generic and the default dictionary for each entity. See extract_entities documentation for details.

Details

This function uses a combination of regular expressions, rule-based approaches, and dictionaries to identify various drug entities of interest. Specific medications to be found are specified with drug_names, which is not case-sensitive or space-sensitive (e.g., 'lamotrigine XR' is treated the same as 'lamotrigineXR'). Entities to be extracted include drug name, strength, dose amount, dose, frequency, intake time, route, duration, and time of last dose. See extract_entities and extract_lastdose for more details.

When searching for medication names of interest, fuzzy matching may be used. The max_dist argument determines the maximum edit distance allowed for such matches. If using fuzzy matching, any drug name with less than 5 characters will only allow an edit distance of 1, regardless of the value of max_dist.

The purpose of the drug_list argument is to reduce false positives by removing information that is likely to be related to a competing drug, not our drug of interest, By default, this is "rxnorm" which calls data(rxnorm_druglist). A custom drug list in the form of a character string can be supplied instead, or can be appended to rxnorm_druglist by specifying drug_list = c("rxnorm", custom_drug_list). medExtractR then uses this list to truncate the search window at the first appearance of an unrelated drug name. This uses publicly available data courtesy of the U.S. National Library of Medicine (NLM), National Institutes of Health, Department of Health and Human

Services; NLM is not responsible for the product and does not endorse or recommend this or any other product. See `rxnorm_druglist` documentation for details.

Most medication entities are searched for in a window after the drug. The dose change entity, or presence of a keyword to indicate a non-current drug regimen, may occur before the drug name. The `flag_window` argument adjusts the width of the pre-drug window. Both `flag_window` and `dosechange_dict` are not default arguments to the extended function `medExtractR_tapering` since that extension uses a more flexible search window and extraction procedure. In the tapering extension, entity extraction is more flexible, and any entity can be extracted either before or after the drug mention. Thus functionality for dose change identification is identical to all other dictionary-based entities.

The `stength_sep` argument is `NULL` by default, but can be used to identify shorthand for morning and evening doses. For example, consider the phrase 'Lamotrigine 300-200' (meaning 300 mg in the morning and 200 mg in the evening). The argument `strength_sep = '-'` identifies the full expression *300-200* as *dose strength* in this phrase.

Value

data.frame with entity information. Only extractions from found entities are returned. If no dosing information for the drug of interest is found, the following output will be returned:

entity	expr	pos
NA	NA	NA

The "entity" column of the output contains the formatted label for that entity, according to the following mapping.
drug name: "DrugName"
strength: "Strength"
dose amount: "DoseAmt"
dose strength: "DoseStrength"
frequency: "Frequency"
intake time: "IntakeTime"
duration: "Duration"
route: "Route"
dose change: "DoseChange"
time of last dose: "LastDose"
Sample output:

entity	expr	pos
DoseChange	decrease	66:74
DrugName	Prograf	78:85
Strength	2 mg	86:90
DoseAmt	1	91:92
Route	by mouth	100:108
Frequency	bid	109:112
LastDose	2100	129:133

References

Nelson SJ, Zeng K, Kilbourne J, Powell T, Moore R. Normalized names for clinical drugs: RxNorm at 6 years. J Am Med Inform Assoc. 2011 Jul-Aug;18(4)441-8. doi: 10.1136/amiajnl-2011-000116. Epub 2011 Apr 21. PubMed PMID: 21515544; PubMed Central PMCID: PMC3128404.

Examples

```
note1 <- "Progrf Oral Capsule 1 mg 3 capsules by mouth twice a day - last
dose at 10pm"
medExtractR(note1, c("prograf", "tacrolimus"), 60, "mg", 2, lastdose=TRUE)
note2 <- "Currently on lamotrigine 150-200, but will increase to lamotrigine 200mg bid"
medExtractR(note2, c("lamotrigine", "ltg"), 130, "mg", 1, strength_sep = "-")
```

---

medExtractR_tapering *Extract Medication Entities From Clinical Note - Extension of* medExtractR *for Tapering applications*

Description

This function identifies medication entities of interest and returns found expressions with start and stop positions.

Usage

```
medExtractR_tapering(
  note,
  drug_names,
  unit,
  max_dist = 0,
  drug_list = "rxnorm",
  lastdose = FALSE,
  strength_sep = NULL,
  ...
)
```

Arguments

note	Text to search.
drug_names	Vector of drug names of interest to locate.
unit	Strength unit to look for (e.g., 'mg').
max_dist	Numeric - edit distance to use when searching for drug_names.
drug_list	Vector of known drugs that may end search window. By default calls rxnorm_druglist. Can be supplemented with expressions in addl_expr.
lastdose	Logical - whether or not last dose time entity should be extracted. See 'Details' section below for more information.

strength_sep	Delimiter for contiguous medication strengths (e.g., '-' for "LTG 200-300").
...	Parameter settings used in dictionary-based entities. For each dictionary-based entity, the user can supply the optional arguments <entity>_fun and <entity>_dict to provide custom extraction functions and dictionaries, respectively. If no additional arguments are provided, medExtractR_tapering will use extract_generic and the default dictionary for each entity. See extract_entities_tapering documentation for details.

Details

This function uses a combination of regular expressions, rule-based approaches, and dictionaries to identify various drug entities of interest, with a particular focus on drugs administered with a tapering schedule. Specific medications to be found are specified with drug_names, which is not case-sensitive or space-sensitive (e.g., 'lamotrigine XR' is treated the same as 'lamotrigineXR'). Entities to be extracted include drug name, strength, dose amount, dose strength, frequency, intake time, route, duration, dose schedule, time keyword, preposition, transition, dispense amount, refill, and time of last dose. While it is still an optional entity in medExtractR_tapering, if lastdose=TRUE then medExtractR_tapering will search for time of last dose in the same search window used for all other entities. As a result, there is no need for the lastdose_window_ext argument. See extract_entities_tapering and extract_lastdose for more details.

When searching for medication names of interest, fuzzy matching may be used. The max_dist argument determines the maximum edit distance allowed for such matches. If using fuzzy matching, any drug name with less than 7 characters will force an exact match, regardless of the value of max_dist. The default value of 7 was selected based on a set of training notes for the drug prednisone, and differs slightly from the default values of 5 for medExtractR. The tapering extension does not use the window_length argument to define the search window, since tapering schedules can be much longer than a static regimens. Instead, medExtractR_tapering dynamically generates the search window based on competing drug names or phrases, and the distance between consecutive entities. The stength_sep argument is NULL by default, and operates in the same manner as it does in medExtractR.

By default, the drug_list argument is "rxnorm" which calls data(rxnorm_druglist). A custom drug list in the form of a character string can be supplied instead, or can be appended to rxnorm_druglist by specifying drug_list = c("rxnorm", custom_drug_list). This uses publicly available data courtesy of the U.S. National Library of Medicine (NLM), National Institutes of Health, Department of Health and Human Services; NLM is not responsible for the product and does not endorse or recommend this or any other product. See rxnorm_druglist documentation for details.

Value

data.frame with entity information. If no dosing information for the drug of interest is found, the following output will be returned:

entity	expr	pos
NA	NA	NA

The "entity" column of the output contains the formatted label for that entity, according to the following mapping.

drug name: “DrugName”
strength: “Strength”
dose amount: “DoseAmt”
dose strength: “DoseStrength”
frequency: “Frequency”
intake time: “IntakeTime”
duration: “Duration”
route: “Route”
dose change: “DoseChange”
dose schedule: “DoseScheule”
time keyword: “TimeKeyword”
transition: “Transition”
preposition: “Preposition”
dispense amount: “DispenseAmt”
refill: “Refill”
time of last dose: “LastDose”
Sample output:

entity	expr	pos
DoseChange	decrease	66:74
DrugName	Prograf	78:85
Strength	2 mg	86:90
DoseAmt	1	91:92
Frequency	bid	101:104
LastDose	2100	121:125

References

Nelson SJ, Zeng K, Kilbourne J, Powell T, Moore R. Normalized names for clinical drugs: RxNorm at 6 years. J Am Med Inform Assoc. 2011 Jul-Aug;18(4)441-8. doi: 10.1136/amiajnl-2011-000116. Epub 2011 Apr 21. PubMed PMID: 21515544; PubMed Central PMCID: PMC3128404.

`preposition_vals` *Keywords Specifying Preposition*

Description

A dictionary with preposition expressions. Such expressions often represent a relationship with an adjacent entity. Since most expressions in this dictionary are very short, we require word boundaries (any character other than a letter or number) to appear on either side of the expression. Example expressions include "for", "to", "until", and "in".

Usage

```
preposition_vals
```

Format

A data frame with preposition expressions (exact and/or regular expressions).

expr A character vector, expressions to consider as preposition.

Examples

```
data(preposition_vals)
```

---

route_vals *Keywords Specifying Route*

Description

A dictionary mapping route expressions to standardized forms, specifying the way in which a medication is administered. Example expressions include "oral", "topical", "IV", and "intravenous".

Usage

```
route_vals
```

Format

A data frame with route expressions (exact and/or regular expressions).

expr A character vector, expressions to consider as route.

value A standardized version of the raw expression. For example, "orally" and "by mouth" both have the standardized form "orally".

Examples

```
data(route_vals)
```

---

rxnorm_druglist *List of Medications*

Description

A dictionary that contains a vector of medication names, primarily derived from RxNorm.

Usage

```
rxnorm_druglist
```

Format

A vector with character strings for competing drug names.

Details

RxNorm is provided by the U.S. National Library of Medicine. This dictionary uses the February 1, 2021 RxNorm files directly downloaded from https://www.nlm.nih.gov/research/umls/rxnorm/docs/rxnormfiles.html.

This list contains ingredient and brand names, cleaned to remove expressions that likely are ambiguous (e.g., 'today' or 'date'). This product uses publicly available data courtesy of the U.S. National Library of Medicine (NLM), National Institutes of Health, Department of Health and Human Services; NLM is not responsible for the product and does not endorse or recommend this or any other product.

References

Nelson SJ, Zeng K, Kilbourne J, Powell T, Moore R. Normalized names for clinical drugs: RxNorm at 6 years. J Am Med Inform Assoc. 2011 Jul-Aug;18(4)441-8. doi: 10.1136/amiajnl-2011-000116. Epub 2011 Apr 21. PubMed PMID: 21515544; PubMed Central PMCID: PMC3128404.

Examples

```
data(rxnorm_druglist)
```

---

string_counts *Counts Strings in Text*

Description

This function counts occurrences of text within one or more phrases.

Usage

```
string_counts(strings, search_data, ignore.case = TRUE)
```

Arguments

strings	character vector; value(s) to find
search_data	character vector; phrase(s) where values may exist
ignore.case	logical; indicates if spelling case matters, defaulting to 'TRUE'

Value

list with two elements; 'cntByTotal' contains total occurrences and 'cntByData' contains occurrences for each element in 'search_data'

Examples

```
note1 <- "I am the very model of a modern major general
I've information vegetable, animal, and mineral
I know the kings of England, and I quote the fights historical
From marathon to Waterloo in order categorical;
I'm very well acquainted, too, with matters mathematical,
I understand equations both the simple and quadratical
About binomial theorem I'm teeming with a lot o' news,
With many cheerful facts about the square of the hypotenuse"
note2 <- "The quick brown fox jumps over the lazy dog"
string_counts(c('I','the','couth'), c(note1, note2))
```

---

string_occurs *Find Strings in Text*

Description

This function searches for text within one or more phrases. Text to look for will be grouped into values that are found and not found.

Usage

```
string_occurs(dict_list, haystack, ignore.case = TRUE, nClust = 2)
```

Arguments

dict_list	character vector; value(s) to find
haystack	character vector; phrase(s) where values may exist
ignore.case	logical; indicates if spelling case matters, defaulting to 'TRUE'
nClust	Number of CPU cores to use, if available. This requires the 'parallel' package.

Value

list with two elements, 'TRUE' and 'FALSE', representing values that are found or not found within the phrase to search.

Examples

```
note1 <- "I am the very model of a modern major general
I've information vegetable, animal, and mineral
I know the kings of England, and I quote the fights historical
From marathon to Waterloo in order categorical;
I'm very well acquainted, too, with matters mathematical,
I understand equations both the simple and quadratical
About binomial theorem I'm teeming with a lot o' news,
With many cheerful facts about the square of the hypotenuse"
note2 <- "The quick brown fox jumps over the lazy dog"
string_occurs(c('kings','quick','couth','brown'), c(note1, note2))
```

string_suggestions *Find Strings and Suggest Misspellings*

Description

This function searches for text within one or more phrases, and looks for partial matches. An exact match of the text should be found in order for a suggestion to made.

Usage

```
string_suggestions(strings, search_data, max_dist = 2, ignore.case = TRUE)
```

Arguments

strings	character vector; value(s) to find
search_data	character vector; phrase(s) where values may exist
max_dist	numeric; edit distance to use for partial matches. The default value is 2.
ignore.case	logical; indicates if spelling case matters, defaulting to 'TRUE'

Value

data.frame with two columns, 'suggestion' and 'match'

Examples

```
string_suggestions('penicillin', 'penicillan, penicillin, or penicilin?')
```

timekeyword_vals *Keywords Specifying Time Keyword*

Description

A dictionary with time keyword expressions representing whether the dosing regimen is past, current, or future. Example expressions include "currently", "remain","not taking", "yesterday", and "past".

Usage

```
timekeyword_vals
```

Format

A data frame with time keyword expressions (exact and/or regular expressions).

expr A character vector, expressions to consider as time keyword.

Examples

```
data(timekeyword_vals)
```

`time_regex` *Keywords Specifying Time Expressions*

Description

A vector of regular expressions to identify different forms of time expressions for last dose time. These are the default values used in `link{extract_lastdose}`.

Usage

```
time_regex
```

Format

A vector with 5 regular expressions for the following categories.

am/pm Time is indicated by the presence of 'am' or 'pm' following a numeric expression.

military Time is given in military time, for unambiguous times of 13:00-23:59.

qualifier_after Am/pm indication is implicit through a qualifying term like 'last night' or 'this morning'. The qualifier occurs after the time, e.g., '10 last night.'

qualifier_before Am/pm indication is implicit through a qualifying term like 'last night' or 'this morning'. The qualifier occurs before the time, e.g., 'last night at 10.'

duration Time (in hours) between the last dose and most recent lab value

Details

Certain expressions which might be considered ambiguous are excluded from the regular expressions presented here. For instance, expressions such as '600' could refer to either 6am or 6pm.

Examples

```
data(time_regex)
```

`transition_vals` *Keywords/Symbols Specifying Transition*

Description

A dictionary with transition symbols and expressions representing a break between consecutive doses within a tapering regimen. This dictionary includes the expressions "then" and "followed by", as well as the punctuation ",(?!\\s?then)" or ";(?!\\s?then)" (i.e., a comma or semicolon not followed by the word "then").

Usage

```
transition_vals
```

Format

A data frame with transition expressions (exact and/or regular expressions).

expr A character vector, expressions to consider as transitions.

Examples

```
data(transition_vals)
```

Index

∗ **datasets**
    addl_expr, 3
    dosechange_vals, 3
    doseschedule_vals, 4
    duration_vals, 4
    frequency_vals, 12
    intaketime_vals, 13
    preposition_vals, 18
    route_vals, 19
    rxnorm_druglist, 19
    time_regex, 23
    timekeyword_vals, 22
    transition_vals, 24
_PACKAGE *(*medExtractR-package*)*, 2

addl_expr, 3, *16*

dosechange_vals, 3
doseschedule_vals, 4
duration_vals, 4

extract_entities, 5, *7*, *9, 10*, *12*, *14*
extract_entities_tapering, 7, *17*
extract_generic, *6*, *9*, 10, *14*, *17*
extract_lastdose, 11, *14*, *17*

frequency_vals, 12

intaketime_vals, 13

medExtractR, *2, 3*, *5, 6*, *9–11*, 13, *16, 17*
medExtractR-package, 2
medExtractR_tapering, *3*, *6, 7*, *9, 10*, *15*, 16

preposition_vals, 18

route_vals, 19
rxnorm_druglist, *14*, *16*, 19

string_counts, 20
string_occurs, 21
string_suggestions, 22

time_regex, *11*, 23
timekeyword_vals, 22
transition_vals, 24