Package 'emend'

July 22, 2025

Title Cleaning Text Data with an AI Assistant

Version 0.1.0

Description Provides functions to clean and standardize messy data, including textual categories and free-text addresses, using Large Language Models. The package corrects typos, expands abbreviations, and maps inconsistent entries to standardized values. Ideal for Bioinformatics, business, and general data cleaning tasks.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Imports cli, jsonlite, rlang, ellmer

Depends R (>= 4.1.0)

LazyData true

Suggests rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

URL https://github.com/anuopensci/emend, https://anuopensci.github.io/emend/

BugReports https://github.com/anuopensci/emend/issues

NeedsCompilation no

Author Emi Tanaka [aut, cph] (ORCID: <https://orcid.org/0000-0002-1455-259X>), Jiajia Li [aut, cre] (ORCID: <https://orcid.org/0009-0003-7143-9336>)

Maintainer Jiajia Li <lijia970324@gmail.com>

Repository CRAN

Date/Publication 2025-04-01 16:20:02 UTC

Contents

`airbnb_listings` *Airbnb listings and reviews*

Description

A sample dataset of Airbnb listings and reviews of properties from Sydney, Australia.

Usage

```
airbnb_listings

airbnb_reviews
```

Format

`airbnb_listings`**:**
A data.frame with 1623 rows and 68 columns

id Airbnb's unique identifier for the listing.

name Name of the listing.

description Detailed description of the listing.

neighborhood_overview Host's description of the neighbourhood.

picture_url URL to the Airbnb hosted regular sized image for the listing.

host_id Airbnb's unique identifier for the host/user.

host_name Name of the host. Usually just the first name(s).

host_since The date the host/user was created. For hosts that are Airbnb guests this could be the date they registered as a guest.

host_location The host's self reported location.

host_about Description about the host.

host_response_time The time interval between when a host responds to an inquiry from a guest.

host_response_rate Percentage of inquiries from potential guests that are responded to by hosts.

host_acceptance_rate That rate at which a host accepts booking requests.

host_is_superhost Whether the host is a super host or not.

host_thumbnail_url A thumbnail of the host.

host_picture_url A URL to the picture of the host.

host_neighbourhood The host neighbourhood.

host_listings_count The number of listings the host has.

host_total_listings_count The number of listings the host has.

host_verifications Host communication verifications.

host_has_profile_pic Whether the host has a profile pic.

host_identity_verified Whether the host has their identity verified.

neighbourhood_cleansed The neighbourhood as geocoded using the latitude and longitude against neighborhoods as defined by open or public digital shapefiles.

latitude Uses the World Geodetic System (WGS84) projection for latitude and longitude.

longitude Uses the World Geodetic System (WGS84) projection for latitude and longitude.

property_type Self selected property type. Hotels and Bed and Breakfasts are described as such by their hosts in this field.

room_type Entire home/apt, Private room, Shared room, or Hotel. Entire places are best if you're seeking a home away from home. With an entire place, you'll have the whole space to yourself. This usually includes a bedroom, a bathroom, a kitchen, and a separate, dedicated entrance. Hosts should note in the description if they'll be on the property or not (ex: "Host occupies first floor of the home"), and provide further details on the listing. Private rooms are great for when you prefer a little privacy, and still value a local connection. When you book a private room, you'll have your own private room for sleeping and may share some spaces with others. You might need to walk through indoor spaces that another host or guest may occupy to get to your room. Shared rooms are for when you don't mind sharing a space with others. When you book a shared room, you'll be sleeping in a space that is shared with others and share the entire space with other people. Shared rooms are popular among flexible travelers looking for new friends and budget-friendly stays.

accommodates The maximum capacity of the listing.

bathrooms The number of bathrooms in the listing.

bathrooms_text The text of the number of bathsroom in the listings.

bedrooms The number of bedrooms.

beds The number of bed(s).

amenities The amenities.

price Daily price in local currency.

minimum_nights Minimum number of night stay for the listing.

maximum_nights Maximum number of night stay for the listing.

minimum_minimum_nights The smallest minimum_night value from the calender (looking 365 nights in the future).

maximum_minimum_nights The largest minimum_night value from the calender (looking 365 nights in the future).

minimum_maximum_nights The smallest maximum_night value from the calender (looking 365 nights in the future).

maximum_maximum_nights The largest maximum_night value from the calender (looking 365 nights in the future).

minimum_nights_avg_ntm The average minimum_night value from the calender (looking 365 nights in the future).

maximum_nights_avg_ntm The average maximum_night value from the calender (looking 365 nights in the future).

has_availability Whether there is availability or not.

availability_30 The availability of the listing x days in the future as determined by the calendar. Note a listing may not be available because it has been booked by a guest or blocked by the host.

availability_60

availability_90

availability_365

number_of_reviews The number of reviews the listing has.

number_of_reviews_ltm The number of reviews the listing has (in the last 12 months).

number_of_reviews_l30d The number of reviews the listing has (in the last 30 days).

first_review The date of the first/oldest review.

last_review The date of the last/newest review.

review_scores_rating The review score for ratings of the listing.

review_scores_accuracy The review score for accuracy of the listing.

review_scores_cleanliness The review score for cleanliness of the listing.

review_scores_checkin The review score for checkin experience of the listing.

review_scores_communication The review score for communication of the listing.

review_scores_location The review score for location of the listing.

review_scores_value The review score for value of the listing.

license The licence/permit/registration number.

instant_bookable Whether the guest can automatically book the listing without the host requiring to accept their booking request. An indicator of a commercial listing.

calculated_host_listings_count The number of listings the host has in the current scrape, in the city/region geography.

calculated_host_listings_count_entire_homes The number of Entire home/apt listings the host has in the current scrape, in the city/region geography.

calculated_host_listings_count_private_rooms The number of Private room listings the host has in the current scrape, in the city/region geography.

calculated_host_listings_count_shared_rooms The number of Shared room listings the host has in the current scrape, in the city/region geography.

reviews_per_month The average number of reviews per month the listing has over the lifetime of the listing.

`airbnb_reviews`**:**

A data.frame with 5679 rows and 6 columns

listing_id Unique identifier for the listing

id Unique identifier for the review

date Date of the review

reviewer_id Unique identifier for the reviewer

reviewer_name Name of the reviewer

comments Text of the review

Source

https://insideairbnb.com/get-the-data/

`alcohol` *Alcohol warehouse and retail sales*

Description

year Year

month Month

supplier Supplier

item_code Item code

item_description Item description

item_type Item type

retail_sales Retail sales

retail_transfers Retail transfers

warehouse_sales Warehouse sales

Usage

```
alcohol
```

Format

An object of class `spec_tbl_df` (inherits from `tbl_df`, `tbl`, `data.frame`) with 1000 rows and 9 columns.

Source

https://catalog.data.gov/dataset/warehouse-and-retail-sales

consumer *Amazon consumer reviews*

Description

A sample of reviews from Amazon in the applicances category.

Usage

```
consumer
```

Format

A data.frame with 4549 rows and 11 columns

overall Overall rating of the product.

verified Whether the reviewer is verified or not.

review_date Review date.

review_id A unique identifier of the reviewer.

reviewer_name The name of the reviewer.

review_text The text of the review.

summary Summary of the review.

vote_helpful The number of helpful votes of the review.

image The images that the reviewer post after they receive the product.

Source

Jianmo Ni, Jiacheng Li, Julian McAuley (2019) Justifying recommendations using distantly-labeled reviews and fined-grained aspects. Empirical Methods in Natural Language Processing .

emend_clean_address *Standardise address format*

Description

This function standardise inconsistent address formats to a standard format.

Usage

```
emend_clean_address(address_vector, chat = get_default_chat())
```

Arguments

address_vector	A character vector that is assumed to be addresses.
chat	A chat object defined by ellmer.

Value

A character vector with converted addresses.

Examples

```
# Convert a vector of inconsistent formatted address to a standard format
options(ellmer_timeout_s = 3600)
x <- c("154 university avenue, acton act 2601",
       "76/2 Cape Street, Dickson ACT 2602",
       "Shop 4/96 Bunda St, Canberra ACT 2601",
       "11 E Row, Canberra ACT 2601",
       "173/46 Macquarie St, Barton ACT 2600",
       "Unit 189/260 City walk, Canberra ACT 2601",
       "the kebab place",
       "i don't know the address")
chat <- ellmer::chat_ollama(model = "llama3.1:8b", seed = 0, echo = "none")
emend_clean_address(x, chat = chat)
```

---

emend_clean_date *Standardise date format*

Description

This function standardise inconsistent date formats.

Usage

```
emend_clean_date(dates_vector, chat = get_default_chat())
```

Arguments

dates_vector	A character vector that is assumed to be dates.
chat	A chat object defined by ellmer.

Value

A vector of Date objects.

Examples

```
x <- c("16/02/1997", "20 November 2024", "24 Mar 2022", "2000-01-01", "Jason",
       "Dec 25, 2030", "11/05/2024", "March 10, 1999")
chat <- ellmer::chat_ollama(model = "llama3.1:8b", seed = 0, echo = "none")
emend_clean_date(x, chat = chat)
```

emend_fct_match *Match input factor to specified levels.*

Description

Match input factor to specified levels.

Usage

```
emend_fct_match(.f, levels = NULL, chat = get_default_chat())
```

Arguments

.f	A factor.
levels	The levels of the factor
chat	A chat object defined by ellmer.

Value

A factor with levels matching the provided `levels` argument.

Examples

```
chat <- ellmer::chat_ollama(model = "llama3.1:8b", seed = 0, echo = "none")
emend_fct_match(messy$country, levels = c("UK", "USA", "Canada", "Australia", "NZ"), chat = chat)
```

emend_fct_reorder *Reorder the levels of the input factor in a meaningful way.*

Description

Reorder the levels of the input factor in a meaningful way.

Usage

```
emend_fct_reorder(.f, chat = get_default_chat())
```

Arguments

.f	A vector of characters or a factor.
chat	A chat object defined by ellmer.

Value

A factor with standardized category labels.

Examples

```
chat <- ellmer::chat_ollama(model = "llama3.1:8b", seed = 0, echo = "none")
emend_fct_reorder(likerts$likert1, chat = chat) |> levels()
```

`emend_lvl_match` *Match the input factor to supplied levels.*

Description

Match the input factor to supplied levels.

Usage

```
emend_lvl_match(.f, levels = NULL, chat = get_default_chat())
```

Arguments

`.f`	A vector of characters or a factor.
`levels`	The levels of the factor.
`chat`	The chat object defined by ellmer.

Value

A named character vector of standardised category labels, with the class `"emend_lvl_match"`. The names correspond to the original messy categories, and the values are the cleaned versions.

Examples

```
chat <- ellmer::chat_ollama(model = "llama3.1:8b", seed = 0, echo = "none")
emend_lvl_match(messy$country,
                levels = c("Asia", "Europe", "North America", "Oceania",
                           "South America"),
                chat = chat)
```

emend_lvl_unique *Get the unique levels of messy categorical data*

Description

The returned value is a vector. The LLM will return full names instead of abbreviations. You can use this functions to clean up your categorical data and obtain unique levels. Double check if the output from LLM is true to your data. This function is generally suitable for categories, not working well with sentences and too many categories.

Usage

```
emend_lvl_unique(.f, chat = get_default_chat())
```

Arguments

.f	A vector of characters or a factor.
chat	A chat object defined by ellmer.

Value

A character vector of standardised category names.

Examples

```
options(ellmer_timeout_s = 3600)
chat <- ellmer::chat_ollama(model = "llama3.1:8b", seed = 0, echo = "none")
emend_lvl_unique(messy$country, chat = chat)
```

emend_translate *Translate text from one language to another.*

Description

Translate text from one language to another.

Usage

```
emend_translate(text, to = "English", chat = get_default_chat())
```

Arguments

text	The text to translate.
to	The language to translate to. The default is "English".
chat	An ellmer Chat object.

Value

A character vector of translated text.

Examples

```
chat <- ellmer::chat_ollama(model = "llama3.1:8b", seed = 0, echo = "none")
emend_translate(c("\u733F\u3082\u6728\u304B\u3089\u843D\u3061\u308B",
                  "\u4F60\u597D", "bon appetit"), chat = chat)
```

`emend_what_language` *Identify the language in the text.*

Description

Identify the language in the text.

Usage

```
emend_what_language(text, chat = get_default_chat())
```

Arguments

`text`	A string or a factor that contains text information.
`chat`	A chat object defined by ellmer

Value

A character vector of language names.

Examples

```
chat <- ellmer::chat_ollama(model = "llama3.1:8b", seed = 0, echo = "none")
emend_what_language(c("\u733F\u3082\u6728\u304B\u3089\u843D\u3061\u308B",
                      "\u4F60\u597D", "bon appetit"), chat = chat)
```

get_default_chat *Get or create the default chat object*

Description

Get or create the default chat object

Usage

```
get_default_chat()
```

Value

A Chat object.

hotel *Hotel reviews*

Description

A sample review of hotels.

Usage

```
hotel
```

Format

A data.frame with 13,193 rows and 8 columns

reviewer_name The name of the reviewer.

reviewer_nationality The nationality of the reviewer.

reviewer_rating Reviewer's rating.

review_date Date of the review.

review_title Title of the review.

review_text Text of the review.

hotel_name Name of the hotel being reviewed.

avg_rating Average rating of the hotel.

Source

https://www.kaggle.com/datasets/nikitaryabukhin/reviewshotel

likerts *A collection of different likert scales*

Description

A data set containing 9 different likert scales.

Usage

```
likerts
```

Format

A data.frame with 40 rows and 9 columns

likert1 A 7-point agreeableness likert scale.

likert2 A 5-point agreeableness likert scale.

likert3 A 5-point agreeableness likert scale as a sentence.

likert4 A 5-point frequency likert scale.

likert5 A 5-point rating likert scale.

likert6 A 5-point likelihood likert scale.

likert7 A 5-point likert scale.

likert8 A 5-point satisfaction likert scale.

likert9 A 6-point priority likert scale.

messy *A collection of messy inputs*

Description

A synthetic dataset that contains inputs with some common standardisation issues.

Usage

```
messy
```

Format

A list of 3 character vectors

country A character vector of countries.

suburb A character vector of suburbs in Australia with various typos.

school A character vector of schools or college (with typos) at the Australian National University.

recipes *Recipes*

Description

A sample of 200 recipes.

Usage

```
recipes
```

Format

A data.frame with 200 rows and 7 columns.

name The name of the dish.

ingredients The list of ingredients.

url The URL of the recipe.

image An image of the dish.

cook_time Cooking time.

prep_time Preperation time.

servings The number of servings in text.

Source

https://github.com/jakevdp/open-recipe-data/tree/main

registration *Registration data for workshops*

Description

A dataset containing registration information for a series of workshops The dataset includes columns for workshop name, affiliation, and the cleaned affiliation by hand.

Usage

```
registration
```

Format

A data frame with 221 rows and 3 columns:

Workshop Character: Workshop name and time.

Affiliation Character: Affiliation input by participants.

GroundTruth Character: The cleaned affiliation type, cleaned manually by author.

Source

Real registration data from the Biological Data Science Institute, Australian National University.

`restaurant` *Review of Iori restaurant*

Description

Iori is a Japanese restaurant in Canberra, Australia. The data contains a sample of 20 reviews from Google maps.

Usage

```
restaurant
```

Format

A data.frame with 20 rows and 2 columns

review The text of the review.

rating A rating out of 5.

`salary` *Ask A Manager Salary Survey 2021*

Description

Ask A Manager Salary Survey 2021

Usage

```
salary
```

Format

A data.frame with 28,083 rows and 18 columns

timestamp The timestampe of response.

age Age cateogry. "How old are you?"

industry Categorical but respondents can enter text for Other. "What industry do you work in?"

job_title Text entry. "Job title"

job_title_context Text entry. "If your job title needs additional context, please clarify here:"

salary_annual Text entry. "What is your annual salary? (You'll indicate the currency in a later question. If you are part-time or hourly, please enter an annualized equivalent – what you would earn if you worked the job 40 hours a week, 52 weeks a year.)"

salarly_additional Text entry. "How much additional monetary compensation do you get, if any (for example, bonuses or overtime in an average year)? Please only include monetary compensation here, not the value of benefits."

currency Categorical entry. "Please indicate the currency"

currency_other Text entry. "If "Other," please indicate the currency here:"

salary_context Text entry. "If your income needs additional context, please provide it here:"

country Text entry. "What country do you work in?"

state Categorical entry. "If you're in the U.S., what state do you work in?"

city Text entry. "What city do you work in?"

experience_overall Categorical entry. "How many years of professional work experience do you have overall?"

experience_in_field Categorical entry. "How many years of professional work experience do you have in your field?"

education Categorical entry."What is your highest level of education completed?"

gender Categorical entry. "What is your gender?"

race Multiselect entry. "What is your race? (Choose all that apply.)"

Source

https://www.askamanager.org/2021/04/how-much-money-do-you-make-4.html

Index