# Package ‘multicastR’

May 12, 2019

**Type** Package

**Title** A Companion to the Multi-CAST Collection

**Version** 1.1.0

**URL** https://multicast.aspra.uni-bamberg.de/

**Description** Provides a basic interface for accessing annotation data from the Multi-CAST collection, a database of spoken natural language texts edited by Geoffrey Haig and Stefan Schnell. The collection draws from a diverse set of languages and has been annotated across multiple levels. Annotation data is downloaded on request from the servers of the University of Bamberg. See the Multi-CAST website <https://multicast.aspra.uni-bamberg.de/> for more information and a list of related publications.

**License** CC BY 4.0

**Encoding** UTF-8

**LazyData** true

**Depends** R (>= 3.0.0), data.table (>= 1.10.0)

**Imports** stringi (>= 1.1.0), xml2 (>= 1.1.0), XML (>= 3.98.0), xtable (>= 1.8.0), gsubfn (>= 0.7)

**RoxygenNote** 6.1.1

**Suggests** testthat

**NeedsCompilation** no

**Author** Nils Norman Schiborr [aut, cre]

**Maintainer** Nils Norman Schiborr <nils-norman.schiborr@uni-bamberg.de>

**Repository** CRAN

**Date/Publication** 2019-05-12 18:00:03 UTC

## R topics documented:

## mcindex *Access the Multi-CAST version index*

### Description

Deprecated with multicastR 1.1.0. Use `mc_index` instead.

### Usage

```
mcindex()
```

## mc_eaf_to_tex *Format object language texts and translations for TeX (WIP)*

### Description

`mc_eaf_to_tex` reads Multi-CAST EAF files and transforms the contents of the `utterance_id`, `utterance`, and `utterance_translation` tiers into a file with LaTeX markup that can be rendered as a multi-column parallel text by the TeX package `paracol`. Regular users of `multicastR` will likely find no use for this function; it's sole purpose is to facilitate the creation of the 'Translated texts' supplementary materials included with Multi-CAST.

### Usage

```
mc_eaf_to_tex(readfrom = getwd(), recursive = FALSE,
  writeto = getwd())
```

### Arguments

| | |
|---|---|
| `readfrom` | Directory from which to read EAF files. Defaults to the current working directory. |
| `recursive` | Logical. If `TRUE`, the function recurses into subdirectories. |
| `writeto` | A directory to which to write output. Defaults to `getwd`. |

### Value

Nothing.

## Examples

```
  ## Not run:
    # read all EAF files in the current working directory,
    # then write TeX files to the same location
    mc_eaf_to_tex()

## End(Not run)
```

---

# mc_eaf_to_tsv *Convert EAF files to TSV (WIP)*

---

## Description

`mc_eaf_to_tsv` converts EAF files produced by the linguistic annotation software ELAN into one or multiple tab-separated values (TSV) tables. The EAF files must have the correct tier structure with the correct tier names, or conversion fails. See the Multi-CAST documentation for details. File are added to the TSV table in the alphabetical order of their file names.

## Usage

```
mc_eaf_to_tsv(readfrom = getwd(), recursive = FALSE, split = FALSE,
  write = FALSE, writeto = getwd(), filename = "")
```

## Arguments

| | |
|---|---|
| `readfrom` | Directory from which to read EAF files. Defaults to the current working directory. |
| `recursive` | Logical. If `TRUE`, the function recurses into subdirectories. |
| `split` | Logical. If `FALSE`, all EAF files that are read are bound into a single data table. If `TRUE`, a list of data tables is returned instead, with one list item per text (which may be split across multiple EAF files). If `write` is `TRUE`, written output is either a single TSV file (for `split == TRUE`) or one TSV file per text read (for `split == FALSE`). In the latter case TSV files combining all texts from each corpus are also produced. |
| `write` | Logical. If `TRUE`, also creates output in `TSV` format. |
| `writeto` | A directory to which to write output. Defaults to `getwd`. Ignored if `write` is `FALSE`. |
| `filename` | A length 1 character vector containing the name of the written output. If empty, defaults to "multicast_YYMM", where 'YY' are the last two digits of the current year and 'MM' the current month. Ignored if `write` is `FALSE` and/or if `split` is `TRUE`, as in the latter case file names are instead generated from text metadata. |

## Value

Either a `data.table` or `list` of `data.table`s of the form produced by `multicast`, containing the annotation values of the EAF files read.

## Examples

```
  ## Not run:
    # read all EAF files in the current working directory,
    # returns a data table of the kind accessed by multicast()
    mc_eaf_to_tsv()

    # also produce a file 'mydata.tsv' containing all read data
    mc_eaf_to_tsv(write = TRUE, filename = "mydata")

    # instead of a single monolithic table, return a list
    # of tables and produce one TSV file for each text
    mc_eaf_to_tsv(write = TRUE, split = TRUE)

## End(Not run)
```

# mc_eaf_to_xml *Convert EAF files to XML (WIP)*

## Description

mc_eaf_to_xml converts EAF files produced by the linguistic annotation software ELAN into one or multiple XML files. The EAF files must have the correct tier structure and names dictated by the Multi-CAST design, else conversion fails. Refer to the Multi-CAST documentation for details about the necessary structure of the EAF files, as well as about the structure of the XML files produced by this function.

## Usage

```
mc_eaf_to_xml(vkey = "", readfrom = getwd(), recursive = FALSE,
  split = FALSE, writeto = getwd(), filename = "",
  skipempty = TRUE)
```

## Arguments

vkey — Character. Version of the annotations. This information is not part of the EAF files, so it needs to be specified manually.

readfrom — Directory from which to read EAF files. Defaults to getwd.

recursive — Logical. If TRUE, the function recurses into subdirectories.

split — Logical. If FALSE, all EAF files that are read are bound into a single XML file. If TRUE, output consists of one XML file for each text read (which may be split across multiple EAF files), plus one XML file bundling all texts from each Multi-CAST corpus. Files combining all texts from each corpus are also produced.

writeto — A directory to which to write output. Defaults to getwd.

| | |
|---|---|
| filename | A length 1 character vector containing the name of the written output. If empty, defaults to "multicast_YYMM", where 'YY' are the last two digits of the current year and 'MM' the current month. Ignored if split is TRUE, as in the latter case file names are instead generated from text metadata. |
| skipempty | Logical. If TRUE, empty leaf nodes in the XML will not be drawn. |

### Examples

```
   ## Not run:
     # read all EAF files in the current working directory
     # and write one XML file for each text to the same
     # location
     mc_eaf_to_xml()

     # same as above, but bundle all data into one large XML file
     # for entire collection plus one XML file for each corpus
     mc_eaf_to_xml(split = TRUE)

## End(Not run)
```

## mc_index *Access the Multi-CAST version index*

### Description

mc_index downloads an index of versions of the Multi-CAST annotation data from the servers of the Language Archive Cologne (LAC) and outputs it as a data.table. The value in the leftmost version column may be passed to the multicast method for access to earlier versions of the annotations.

### Usage

```
mc_index()
```

### Value

A data.table with five columns:

[, 1] version Version key. YYMM format. Used for multicast's vkey argument.

[, 2] date Publication date. YYYY-MM-DD format.

[, 3] size Total file size in kilobytes.

[, 4] texts Number of texts.

[, 5] corpora Names of the corpora (languages) included in the version.

### See Also

multicast.

## Examples

```
  ## Not run:
    # retrieve and print version index
    mc_index()

## End(Not run)
```

# mc_reflist *Convert TSV referent lists to TEX*

## Description

`mc_reflist` reads lists of referents in TSV format and outputs them as files with TEX markup that can be rendered as a multi-column parallel text by the TeX package `paracol`. Regular users of `multicastR` will likely find no use for this function; it's sole purpose is to facilitate the creation of the 'List of referents' supplementary materials included with Multi-CAST.

## Usage

```
mc_reflist(readfrom = getwd(), recursive = FALSE, writeto = getwd())
```

## Arguments

| | |
|---|---|
| `readfrom` | Directory from which to read EAF files. Defaults to `getwd`. |
| `recursive` | Logical. If `TRUE`, the function recurses into subdirectories. |
| `writeto` | A directory to which to write output. Defaults to `getwd`. |

## Value

Nothing.

## Examples

```
  ## Not run:
    # read all TSV files in the current working directory
    # and write one TEX file for each TSV file to the same
    # location
    mc_reflist()

## End(Not run)
```

## mc_table *Generate tables with summarized GRAID counts (WIP)*

### Description

Constructs simple tables with counts for certain combinations of GRAID form, person/animacy, and function symbols. In the current iteration, the GRAID categories counted for the tables are predetermined and cannot be changed by the user. The TEX files that can optionally be written by this function are used for the 'Corpus counts' in the Multi-CAST documentation.

### Usage

```
mc_table(data, by = "all", format = "wide", write = FALSE,
  writeto = getwd(), output = "tex")
```

### Arguments

| | |
|---|---|
| data | A data.table in multicastR format. |
| by | Character. "all" places all data in one table, "corpus" generates one table for each corpus, and "text" one table for each text. |
| format | Unused. Will be used to select between "wide" and "long" table layouts. |
| write | Logical. If TRUE, writes output to file. |
| writeto | A directory to which to write output. Defaults to getwd. Ignored if write is FALSE. |
| output | Unused. Will be used to specify the file format to write as. Currently only output as TEX files is supported. |

### Value

A table.

### Examples

```
 ## Not run:
   # generate a summary table for the entire collection
   mc <- multicast()
   mc_table(mc)

   # generate a summary table for the English corpus
   mc_table(mc[corpus == "english", ])

## End(Not run)
```

## multicast *Access Multi-CAST annotation data*

### Description

multicast downloads the Multi-CAST annotation data from the servers of the University of Bamberg and outputs them as a data.table. As the Multi-CAST collection is amenable to extension by additional data sets and annotation schemes, multicast takes an optional argument to select earlier versions of the annotation data to ensure scientific accountability and reproducability.

### Usage

```
multicast(vkey, legacy.colnames = FALSE)
```

### Arguments

vkey
: A numeric or character vector of length 1 specifying the requested version of the annotation values. Must be one of the four-digit version keys in the first column of mc_index, or empty. If empty or no value is supplied, multicast automatically retrieves the most recent version of the annotations. See the examples below for an illustration.

legacy.colnames
: If TRUE, renames the text and gword columns to what they were called prior to version 1.1.0 of the package (i.e. file, word). This option will be removed in the future.

### Value

A data.table with eleven columns:

[, 1] corpus The name of the corpus.

[, 2] text The title of the text. If legacy.colnames is TRUE, this column is named file instead.

[, 3] uid The utterance identifier. Uniquely identifies an utterance within a text.

[, 4] gword Grammatical words. The tokenized utterances in the object language. If legacy.colnames is TRUE, this column is named word instead.

[, 5] gloss Morphological glosses following the Leipzig Glossing Rules.

[, 6] graid Annotations using the GRAID scheme (Haig & Schnell 2014).

[, 7] gform The form symbol of a GRAID gloss.

[, 8] ganim The person-animacy symbol of a GRAID gloss.

[, 9] gfunc The function symbol of a GRAID gloss.

[, 10] refind Referent tracking using the RefIND scheme (Schiborr et al. 2018).

[, 11] reflex The information status of newly introduced referents, using a simplified version of the RefLex scheme (Riester & Baumann 2017).

### Licensing

### Citing Multi-CAST

Data from the Multi-CAST collection should be cited as:

- Haig, Geoffrey & Schnell, Stefan (eds.). 2015. *Multi-CAST: Multilinguial Corpus of Annotated Spoken Texts*. (https://multicast.aspra.uni-bamberg.de/) (Accessed *date*.)

If for some reason you need to cite this package on its own, please refer to `citation(multicastR)`.

### References

- Haig, Geoffrey & Schnell, Stefan. 2014. *Annotations using GRAID (Grammatical Relations and Animacy in Discourse): Introduction and guidelines for annotators.* Version 7.0. (https://multicast.aspra.uni-bamberg.de/)
- Riester, Arndt & Baumann, Stefan. 2017. The RefLex scheme – Annotation guidelines. *SinSpeC: Working papers of the SFB 732* 14. (https://dx.doi.org/10.18419/opus-9011))
- Schiborr, Nils N. & Schnell, Stefan & Thiele, Hanna. 2018. *RefIND – Referent Indexing in Natural-language Discourse: Annotation guidelines.* Version 1.1. (https://multicast.aspra.uni-bamberg.de/)

### See Also

`mc_index`

### Examples

```
  ## Not run:
    # retrieve and print the most recent version of the
    # Multi-CAST annotations
    multicast()

    # retrieve and print the version of the annotation data
    # published in June 2016
    multicast(1606)   # or: multicast("1606")

## End(Not run)
```

multicastR *multicastR: A companion to the Multi-CAST collection.*

## Description

The multicastR package provides a basic interface for accessing annotation data in the Multi-CAST collection (edited by Geoffrey Haig and Stefan Schnell), a database of spoken natural language texts that draws from a diverse set of languages and has been annotated across multiple levels. Annotation data is downloaded on command from the servers of the University of Bamberg via the multicast method. Details on the Multi-CAST project and a list of publications can be found online at https://multicast.aspra.uni-bamberg.de/.

## Licensing

The Multi-CAST annotation data accessed by the multicast method is published under a *Create Commons Attribution 4.0 International* (CC-BY 4.0) licence (https://creativecommons.org/licenses/by-sa/4.0/). Please refer to the collection documentation for information on how to give proper credit to its contributors.

## Citing Multi-CAST

Data from the Multi-CAST collection should be cited as:

- Haig, Geoffrey & Schnell, Stefan (eds.). 2018[2015]. *Multi-CAST: Multilinguial Corpus of Annotated Spoken Texts*. (http://multicast.aspra.uni-bamberg.de/) (Accessed *date*.)

If for some reason you need to cite this package specifically, please refer to citation(multicastR).

## See Also

multicast, mcindex.

# Index