Package 'microdemic'

October 25, 2018

Type Package

Title 'Microsoft Academic' API Client

Description The 'Microsoft Academic Knowledge' API provides programmatic access to scholarly articles in the 'Microsoft Academic Graph' (<https://academic.microsoft.com/>). Includes methods matching all 'Microsoft Academic' API routes, including search, graph search, text similarity, and interpret natural language query string.

Version 0.4.0

License MIT + file LICENSE

LazyData TRUE

Encoding UTF-8

URL https://github.com/ropensci/microdemic

BugReports https://github.com/ropensci/microdemic/issues

Imports crul (>= 0.5.2), jsonlite (>= 1.5), data.table, tibble

Suggests testthat, vcr

RoxygenNote 6.1.0

NeedsCompilation no

Author Scott Chamberlain [aut, cre] (<https://orcid.org/0000-0003-1444-9135>)

Maintainer Scott Chamberlain <myrmecocystus+r@gmail.com>

Repository CRAN

Date/Publication 2018-10-25 04:30:21 UTC

R topics documented:

`microdemic-package` *Microsoft Academic Client*

Description

Microsoft Academic Client

Links

- web interface: https://academic.microsoft.com/
- API docs: https://dev.labs.cognitive.microsoft.com/docs/services/56332331778daf02acc0a50b/operations/565d9001ca73072048922d97

Package API

- `ma_search()`
- `ma_calchist()`
- `ma_evaluate()`
- `ma_interpret()`
- `ma_similarity()`
- `ma_abstract()`
- `ma_graph_search()`

Authentication

See https://labs.cognitive.microsoft.com/en-us/subscriptions to get an API key. Make sure you enable Academic Knowledge and. They should give you two API keys - use either one. You can always pass your API key as a parameter to functions in `microdemic`, but we strongly encourage you to set an environment variable, named `MICROSOFT_ACADEMIC_KEY`

To set the environment variable for the current R session only, run `Sys.setenv(MICROSOFT_ACADEMIC_KEY = "yourkey")`. Then `microdemic` will pick up this key and you do not have to pass your key as a parameter to the functions.

Even better, save your environment variable in the file that R will use to read in environment variables like `MICROSOFT_ACADEMIC_KEY=yourkey`. This may vary by system and machine. See `?Startup` for details. It's likely your `.Renviron` file that is stored in your home directory. Or could be another environment variable holding file like `.zshrc` or `.bash_profile`. After saving the file, make sure to quit R, source that file like `source .Renviron`, then start R again. Or restart your RStudio session or similar for other R scenarios.

Rate limits

- 10,000 transactions per day
- 3 per second for interpret
- 1 per second for evaluate
- 6 per minute for calcHistogram

Author(s)

Scott Chamberlain <myrmecocystus@gmail.com>

`ma_abstract` *Fetch abstracts*

Description

Fetch abstracts

Usage

```
ma_abstract(query, count = 10, offset = 0, orderby = NULL,
  model = "latest", key = NULL, ...)
```

Arguments

`query`	(character) query terms
`count`	(intger) number of records to return. default: 10
`offset`	(intger) record number to start at. default: 0
`orderby`	(logical) column by which to order. default: none
`model`	(character) Name of the model that you wish to query. One of 'latest' or 'beta-2015'. Default: latest
`key`	(character) microsoft academic API key, see the `Authentication` section in microdemic-package
`...`	curl options passed on to crul::HttpClient

Value

data.frame, with two columns: `Id` and `abstract`

Examples

```
## Not run:
ma_abstract(query = "Y=2010", count = 10)
ma_abstract(query = "Y=[2010, 2012)", count = 10)

## End(Not run)
```

ma_calchist *CalcHistogram API*

Description

CalcHistogram API

Usage

```
ma_calchist(query, count = 10, offset = 0, atts = c("Id", "AA.AuN",
  "J.JN", "Ti", "Y", "E", "CC"), model = "latest", key = NULL, ...)
```

Arguments

query	(character) query terms
count	(intger) number of records to return. default: 10
offset	(intger) record number to start at. default: 0
atts	(character) fields to return, in a character vector. See https://docs.microsoft.com/en-us/azure/cognitive-services/academic-knowledge/entityattributes for details.
model	(character) Name of the model that you wish to query. One of 'latest' or 'beta-2015'. Default: latest
key	(character) microsoft academic API key, see the Authentication section in microdemic-package
...	curl options passed on to crul::HttpClient

Examples

```
## Not run:
res <- ma_calchist(query = "And(Composite(AA.AuN=='jaime teevan'),Y>2012)",
  atts = c('Y', 'F.FN'))
res$histograms$histogram

## End(Not run)
```

ma_evaluate *Evaluate API*

Description

Evaluate API

Usage

```
ma_evaluate(query, count = 10, offset = 0, orderby = NULL,
  atts = c("Id", "AA.AuN", "J.JN", "Ti", "Y", "E", "CC"),
  model = "latest", key = NULL, ...)
```

Arguments

query	(character) query terms
count	(intger) number of records to return. default: 10
offset	(intger) record number to start at. default: 0
orderby	(logical) column by which to order. default: none
atts	(character) fields to return, in a character vector. See https://docs.microsoft.com/en-us/azure/cognitive-services/academic-knowledge/entityattributes for details.
model	(character) Name of the model that you wish to query. One of 'latest' or 'beta-2015'. Default: latest
key	(character) microsoft academic API key, see the Authentication section in microdemic-package
...	curl options passed on to crul::HttpClient

Value

a list of length two, with expr (character) and entities (data.frame)

References

Query expression syntax https://docs.microsoft.com/en-us/azure/cognitive-services/academic-knowledge/queryexpressionsy

Examples

```
## Not run:
# any numeric value that starts with 19
ma_evaluate(query = "Y='19'...")
# author name matching, and request certain fields back
ma_evaluate(query = "Composite(AA.AuN=='smith')",
  atts = c('Id', 'AA.AuN'))
# matching year 2010 and request Id and Y fields
ma_evaluate(query = "Y=2010", atts = c('Id', 'Y'))
# range of years: includes only left boundary value: 2010, 2012
ma_evaluate(query = "Y=[2010, 2012)", atts = c('Id', 'Y'))
# range by specific dates
ma_evaluate(query = "D=['2010-02-03','2010-02-05']", atts = c('Id', 'Y'))
# matching author and matching affiliation
x <- "Composite(And(AA.AuN='mike smith',AA.AfN='harvard university'))"
ma_evaluate(x)

## End(Not run)
```

`ma_graph_search` *Graph search API*

Description

Graph search API

Usage

```
ma_graph_search(query, mode = "json", key = NULL, ...)
```

Arguments

`query`	(character) query terms
`mode`	(character) json (default) or lambda
`key`	(character) microsoft academic API key, see the `Authentication` section in microdemic-package
`...`	curl options passed on to crul::HttpClient

Note

THIS FUNCTION CURRENTLY DOES NOT WORK AS OF 2018-10-22 - IT'S NOT CLEAR HOW TO QUERY AGAINST THIS API ROUTE (`academic/v1.0/graph/search`)

References

graph search method: https://docs.microsoft.com/en-us/azure/cognitive-services/academic-knowledge/graphsearchmethod json search: https://docs.microsoft.com/en-us/azure/cognitive-services/academic-knowledge/jsonsearchsyntax lambda search: https://docs.microsoft.com/en-us/azure/cognitive-services/academic-knowledge/lambdasearchsyntax

Examples

```
## Not run:
# x <- '{
# "path": "/paper/AuthorIDs/author",
#  "paper": {
#    "type": "Paper",
#    "NormalizedTitle": "graph engine",
#    "select": [ "OriginalTitle" ]
#  },
#  "author": {
#    "return": { "type": "Author", "Name": "bin shao" }
#  }
# }'
#
# res <- ma_graph_search(query = x)
# res$Results
```

```
# do.call(rbind, res$Results)

## End(Not run)
```

`ma_interpret` *Interpret API*

Description

Interpret API

Usage

```
ma_interpret(query, count = 10, offset = 0, complete = TRUE,
  key = NULL, ...)
```

Arguments

`query`	(character) query terms
`count`	(intger) number of records to return. default: 10
`offset`	(intger) record number to start at. default: 0
`complete`	(logical) `TRUE` means that auto-completion suggestions are generated based on the grammar and graph data. default: `TRUE`
`key`	(character) microsoft academic API key, see the `Authentication` section in microdemic-package
`...`	curl options passed on to crul::HttpClient

References

https://docs.microsoft.com/en-us/azure/cognitive-services/academic-knowledge/interpretmethod

Examples

```
## Not run:
res <- ma_interpret(query = "papers by jaime'...")
res$query
res$interpretations
res$interpretations$parse
res$interpretations$rules
res$interpretations$rules[[1]]

expr <- res$interpretations$rules[[1]]$output.value
ma_evaluate(expr)

## End(Not run)
```

`ma_search` *Search - higher level method*

Description

Search - higher level method

Usage

```
ma_search(query, count = 10, offset = 0, orderby = NULL,
  atts = c("Id", "AA.AuN", "J.JN", "Ti", "Y", "E", "CC"),
  model = "latest", key = NULL, ...)
```

Arguments

`query`	(character) query terms
`count`	(intger) number of records to return. default: 10
`offset`	(intger) record number to start at. default: 0
`orderby`	(logical) column by which to order. default: none
`atts`	(character) fields to return, in a character vector. See https://docs.microsoft.com/en-us/azure/cognitive-services/academic-knowledge/entityattributes for details.
`model`	(character) Name of the model that you wish to query. One of 'latest' or 'beta-2015'. Default: latest
`key`	(character) microsoft academic API key, see the `Authentication` section in microdemic-package
`...`	curl options passed on to crul::HttpClient

References

https://academic.microsoft.com/ https://azure.microsoft.com/en-us/services/cognitive-services/academic-knowledge/ https://docs.microsoft.com/en-us/azure/cognitive-services/academic-knowledge/home https://westus.dev.cogniti
https://westus.dev.cognitive.microsoft.com/docs/services/56332331778daf02acc0a50b/operations/565d753be597ed16ac3ffc

Examples

```
## Not run:
ma_search(query = "Y='19'...")

## End(Not run)
```

`ma_similarity` *Similarity API*

Description

Similarity API

Usage

```
ma_similarity(s1, s2, method = "GET", model = "latest", key = NULL,
  ...)
```

Arguments

`s1, s2`	(character) strings 1 and 2. required
`method`	(character) one of GET (default) or POST
`model`	(character) Name of the model that you wish to query. One of 'latest' or 'beta-2015'. Default: latest
`key`	(character) microsoft academic API key, see the `Authentication` section in microdemic-package
`...`	curl options passed on to crul::HttpClient

Value

a single value representing the cosine similarity of the text inputs of s1 and s2. The output is represented by a floating point between -1.0 and +1.0. The similarity API evaluates the strings base on their academic concepts, with +1.0 being the most similar and -1.0 being the least similar.

Examples

```
## Not run:
s1 <- "Using complementary priors, we derive a fast greedy algorithm that
can learn deep directed belief networks one layer at a time, provided the
top two layers form an undirected associative memory"

s2 <- "Deepneural nets with a large number of parameters are very powerful
machine learning systems. However, overfitting is a serious problem in
such networks"

ma_similarity(s1, s2)

ma_similarity(s1, s2, method = "POST")

## End(Not run)
```

Index