Package 'rtika'

January 30, 2018

Type Package

Title R Interface to 'Apache Tika'

Version 0.1.1

Author
Sasha Goodman <goodmansasha@gmail.com>[aut], The Apache Software Foundation [aut,cph]

Maintainer Sasha Goodman <goodmansasha@gmail.com>

Suggests sys,
jsonlite,
xml2,
data.table,
curl,
testthat,
knitr,
rmarkdown,
covr

License Apache License 2.0 | file LICENSE

SystemRequirements Java (>=7) | openjdk-7-jre (via apt) | java-1.7.0-openjdk (via yum) | openjdk-8-jre (via apt) | java-1.8.0-openjdk (via yum)

Description Extract text and metadata from over a thousand file types. Get either plain text or structured XHTML. This R interface includes the Tika software. Its source is available at https://github.com/apache/tika.

Depends R (>= 3.1.0)

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

URL http://github.com/predict-r/rtika

BugReports http://github.com/predict-r/rtika/issues

VignetteBuilder knitr

R topics documented:

`tika` *R Interface to 'Apache Tika'*

Description

Extract text and metadata from over a thousand file types. Get either plain text or structured `XHTML`. Metadata includes `Content-Type`, character encoding, and Exif data from jpeg or tiff images. See the supported file types: `https://tika.apache.org/1.17/formats.html`.

Usage

```
tika(input, output = c("text", "jsonRecursive", "xml", "html")[1],
  output_dir = "", java = "java", jar = system.file("java",
  "tika-app-1.17.jar", package = "rtika"), threads = 1, args = character(),
  quiet = TRUE, cleanup = FALSE, lib.loc = .libPaths())
```

Arguments

`input`	Character vector of paths to the input documents. Strings starting with 'http://','https://', or 'ftp://' are downloaded to a temporary directory first. Each file will be analyzed but not changed.
`output`	Optional character vector of the output format. By default, `"text"` gets plain text without metadata. `"xml"` and `"html"` get `XHTML` text with metadata. `"jsonRecursive"` gets `XHTML` text and `json` metadata. `c("jsonRecursive","text")` or `c("J","t")` gets plain text and `json` metadata. See the 'Output Details' section.
`output_dir`	Optional directory path to save the converted files in. Tika may overwrite files so an empty directory is best. See the 'Output Details' section before using.
`java`	Optional command to invoke Java. For example, it could be to the full path of a particular Java version. See the Configuration section below.
`jar`	Optional alternative path to a `tika-app-X.XX.jar`. Useful if this package becomes out of date.
`threads`	Integer of the number of file consumer threads Tika uses. Defaults to 1.
`args`	Optional character vector of additional arguments for Tika, that are not yet implemented in this R interface, in the pattern of `c('-arg1','setting1','-arg2','setting2')`. Currently settable arguments include `-timeoutThresholdMillis` (Number of milliseconds allowed to a parse before the process is killed and restarted), `-maxRestarts` (Maximum number of times the watchdog process will restart the child process), `-includeFilePat` (Regular expression to determine which files to process, e.g. `"(?i)\.pdf"`), `-excludeFilePat`, and `-maxFileSizeBytes`. These are documented in the .jar –help command.

quiet	Logical if Tika command line messages and errors are to be supressed. Defaults to TRUE.
cleanup	Logical to clean up temporary files after running the command, which can accumulate. They are in tempdir(). These files normally be removed at the end of the R session anyhow.
lib.loc	Optional character vector describing the library path(s) containing curl, data.table or sys packages. Normally, it's best to install the packages and leave this parameter alone. The parameter is included mainly for package testing.

Value

A character vector in the same order as the input, and the same lenth. Unprocessed files are NA. See the Output Details section below.

Output Details

If an input file did not exist, could not be downloaded, was a directory, or Tika could not process it, the result will be an NA for that file. This should not disrupt the processing of other files in the input, although there may be warnings if it was Tika's fault.

By default, output = "text" and this produces plain text with no metadata. Some formatting is preserved using tabs, newlines and spaces.

Setting output to either "xml" or the shortcut "x" will produce a strict form of HTML known as XHTML, with metadata in the head node and formatted text in the body. Content retains more formatting with "xml". For example, a Word or Excel table will become a HTML table, with table data as text in td elements. The "html" option and its shortcut "h" seem to produce the same result as "xml". Parse XHTML output with xml2::read_html.

Setting output to "jsonRecursive" or its shortcut "J" produces a tree structure in 'json'. Metadata fields are at the top level. The XHTML or plain text will be found in the X-TIKA:content field. By default the text is XHTML. This can be changed to plain text like this: output=c("jsonRecursive","text") or output=c("J","t"). This syntax is meant to mirror Tika's. Parse json with jsonlite::fromJSON.

If output_dir is specified, then the converted files will also be saved to this directory. It's best to use an empty directory because Tika may overwrite existing files. Tika seems to add an extra file extension to each file to reduce the chance, but it's still best to use an empty directory. The file locations within the output_dir maintain the same general path structure as the input files. Downloaded files have a path similar to the 'tempdir()' that R uses. The original paths are now relative to output_dir. Files are appended with .txt for the default plain text, but can be .json, .xml, or .html depending on the output setting. One way to get a list of the processed files is to use list.files with recursive=TRUE. If output_dir is not specified, files are saved to a volatile temp directory named by tmpdir() and will be deleted when R shuts down. If this function will be run on very large batches repeatedly, these temporary files can be cleaned up every time by adding cleanup=TRUE.

Background

Tika is a foundational library for several Apache projects such as the Apache Solr search engine. It has been in development since at least 2007. The most efficient way I've found to process many thousands of documents is Tika's 'batch' mode, which is the only mode used in 'rtika'. There

are potentially more things that can be done with this package, given enough time and attention, because Apache Tika includes many libraries and methods in its .jar file. The source is available at: https://tika.apache.org/.

Configuration

This package includes the tika-app-X.XX.jar. This jar works with Java 7. Tika in mid-2018 needs Java 8, so it's best to install that version if possible.

By default, this R package internally invokes Java by calling the java command from the command line. To specify the path to a particular Java version, set the path in the java attribute of the tika function.

Other command line arguments can be set with args. See the options for version 1.17 here: https://tika.apache.org/1.17/gettingstarted.html

Having the sys package is suggested but not required. The sys package can dramatically speed up the initial call to Java each time this function is run, which is useful if you are calling this function again and again. Installing sys after rtika will work as well as installing it before. If you find yourself calling tika repeatedly, consider supplying a long character vector of files to input instead of an individual file each time.

Having the data.table package installed will slightly speed up the communication between R and Tika, but especially if there are hundreds of thousands of documents to process.

The curl package downloads files quickly, if the user includes urls in the input. In testing, curl is required on Windows to avoid errors, and more work may still be needed to make Windows parse reliably.

Examples

```
#' #extract text
input= 'https://cran.r-project.org/doc/manuals/r-release/R-data.pdf'
text = tika(input)
cat(substr(text[1],45,450))

#get metadata
if(requireNamespace('jsonlite')){
  json = tika(input,'J') # capital J is shortcut for jsonRecursive

  metadata = jsonlite::fromJSON(json[1])
  str(metadata) #meta meta-data

  metadata$'Content-Type' # [1] "application/pdf"
  metadata$producer # [1] "pdfTeX-1.40.18"
  metadata$'Creation-Date' # [1] "2017-11-30T13:39:02Z"
}
```

Index