The Journal of Open Source Software

daiR: an R package for OCR with Google Document AI

Thomas Hegghammer[*1]

1 Senior Research Fellow, Norwegian Defence Research Establishment (FFI)

DOI: 10.21105/joss.03538

Software

- Review
- Repository
- Archive

Editor: Nikoleta Glynatsi

Reviewers:

- @cjbarrie
- @shivam11

Submitted: 24 June 2021
Published: 26 July 2021

Statement of need

Optical character recognition (OCR) promises to open up centuries worth of text to computational analysis. But OCR software has long been sensitive to visual noise and weak on non-Western languages. In April 2021, Google launched Document AI (DAI), a server-based processor offering high-accuracy OCR for over sixty languages (Vanguri, 2021). The `daiR` (Hegghammer, 2021b) package provides an R interface to the Document AI API along with additional tools for output parsing and visualization.

Summary

Text as data is a growing field in the social sciences and digital humanities, but computational access to text produced before the late 20th century has been limited by the difficulty of extracting text from document scans. Established OCR libraries such as Tesseract (Tesseract, 2021) are highly sensitive to noise and often require extensive corpus-specific adaptations to render text accurately.

The past two years have seen the introduction of server-based OCR processors, such as Amazon Textract (Amazon, 2021) and Google Document AI, which offer very high accuracy out of the box (Hegghammer, 2021a). Of the two, DAI performs better in benchmarking tests and offers broader language support.

In R, where many scholars do their text analysis work, there are packages for Tesseract (Ooms, 2021) and Amazon Textract (Kretch & Banker, 2021), but not for Document AI. The primary objective of `daiR` is therefore to provide access, from within R, to all the main functionalities of the Document AI API. The secondary aim is to offer tools to help parse the output returned by the DAI processor.

DAI is part of Google Cloud Services (GCS), a suite of cloud computing services for storage, analytics, and machine learning. `daiR` joins a family of existing R packages that interface with GCS, such as `googleLanguageR` (Edmondson, 2020), that together allows for the implementation of multiple GCS tools into an R-based text mining workflow.

DAI currently has three API endpoints — v1, v1beta2, and v1beta3 — with partly overlapping sets of methods. Table 1 summarises their features and associated `daiR` functions. (Note that `daiR` does not provide access to v1beta3 as it is very similar to v1beta2.)

`daiR` function	API endpoint	Processing method	Main features
dai_sync()	v1	Synchronous (single)	Text (60+ languages)
dai_async()	v1	Asynchronous (batch)	Text (60+ languages)
dai_sync_tab()	v1beta2	Synchronous (single)	Text (English only), tables
dai_async_tab()	v1beta2	Asynchronous (batch)	Text (English only), tables

*corresponding author

Table 1: Overview of document processing functions in `daiR`

Documents can be processed synchronously or asynchronously. The first method takes a single document and returns the result directly into R. The second allows batch processing and involves uploading documents to a Google Cloud Storage bucket, telling DAI where to find them, and retrieving the JSON file output from the same bucket. The R package `googleCloudStorageR` (Edmondson, 2021) provides an interface with Google Cloud Storage.

Use of Document AI requires a GCS account. The default authentication method in `daiR` is with a service account key file, but users can authenticate in other ways and pass the access token into all of `daiR`'s processing functions. Document AI is a paid service currently priced at $1.50 per 1000 pages.

`daiR` also includes a range of tools to process DAI's output, which comes in complex JSON files. One set of functions extracts text and table data from the JSON files and brings them into R as character vectors or data frames. Another set draws block, paragraph, line, and token boundary boxes on images of the submitted documents, to help with visual inspection. A third group of functions helps rearrange text blocks in the cases where Document AI has misread their order. Document AI has near-perfect character recognition, but its parsing of complex page layouts is fallible. This problem is likely to diminish over time as Document AI's algorithm trains on ever larger document data sets. In the meantime, `daiR` makes it relatively easy to correct DAI's errors and obtain an accurately rendered text.

`daiR` is the first R tool to offer high-accuracy text extraction from noisy historical documents out of the box. Until now, scholars have often dealt with Tesseract's high error rates by treating error as noise and using bag-of-words techniques such as topic modeling. Low-error OCR opens up for wider use of natural language processing and other methods that require correctly parsed and ordered text. DAI's improved language coverage may also help reduce the prevalence of English-language data in computational text analysis.

Acknowledgements

I thank Mark Edmondson, Trond Arne Sørby, Neil Ketchley, and Hallvar Gisnås for contributions to this project.

References

Amazon. (2021). *Amazon Textract: Easily extract printed text, handwriting, and data from virtually any document*. https://aws.amazon.com/textract/

Edmondson, M. (2020). *googleCloudLanguageR: Call Google's 'Natural Language' API, 'Cloud Translation' API, 'Cloud Speech' API and 'Cloud Text-to-Speech' API*. https://cran.r-project.org/package=googleLanguageR

Edmondson, M. (2021). *googleCloudStorageR: Interface with Google Cloud Storage API*. https://cran.r-project.org/package=googleCloudStorageR

Hegghammer, T. (2021a). OCR with Tesseract, Amazon Textract, and Google Document AI: A Benchmarking Experiment. *Socarxiv*. https://doi.org/10.31235/osf.io/6zfvs

Hegghammer, T. (2021b). *daiR: An R package for OCR in Google Document AI*. https://cran.r-project.org/package=daiR

Kretch, M., & Banker, A. (2021). *paws: Amazon Web Services Software Development Kit*. https://cran.r-project.org/package=paws

Ooms, J. (2021). *tesseract: Open Source OCR Engine*. https://cloud.r-project.org/web/packages/tesseract/index.html

Tesseract. (2021). Tesseract OCR. In *GitHub repository*. GitHub. https://github.com/tesseract-ocr/tesseract

Vanguri, S. (2021). Customers cut document processing time and costs with DocAI solutions, now generally available. *Google Cloud Blog*.