# text-extraction-service

TES is a simple Go service for extracting and storing textual content from PDF, RTF and legacy MS Word (.doc) documents.

## Status

This started as an exercise in using Golang and cgo. But it is about to be used in production (at least for PDFs). The use case is the fast processing of binary documents for repeated search machine indexation.

The RegEx-based RTF parser is rather inefficient.

Apache [Tika](https://tika.apache.org/) is definitively a more versatile and mature solution to be considered.

## Features

* Support for PDFs, RTFs and legacy MS Word (.doc) files
* Support for three runtime-pluggable C/C++ PDF engines
  + Google Chromium's [PDFium](https://pdfium.googlesource.com/pdfium/)
  + Free Desktops [Poppler](https://poppler.freedesktop.org/)
  + Artifex' [MuPDF](https://mupdf.com/)
* Optional Dehyphenation of extracted text, specifically for German
* Extraction of document metadata (title, author, creation date etc)
* Store extracted text and metadata in NATS for faster retrieval
* NATS can be embedded or run externally (e.g. as a cluster)
* Support for NATS microservice interface
* (Experimental) Optical character recognition by [Tesseract OCR](https://github.com/tesseract-ocr/) (useful for images containing text and scanned PDFs)

## Unsupported

* Processing local files with the file: transport
* Processing password protected files
* Processing files from web servers that require authentication of any kind (cookie, header, referral, user agent etc)
* A lot of common document formats, including odt, docx, html, xml

## License

This service inherits the Open Source license of the PDF lib used at runtime:

* PDFium: [Apache-2](https://pdfium.googlesource.com/pdfium/+/master/LICENSE)
* Poppler: GPL-2.0
* MuPDF: AGPL-3.0 (commercial license available)

This approach is unusual and probably problematic. There is an unresolved debate about the consequences of linking against GPL libraries concerning the licensing obligations of a software. With the current purego-based implementation there is not even a compile-time dependency on Poppler or MuPDF. But these libraries can still be loaded at runtime and will run in the same address space, just like any other dynamically linked library.

I'm not an OSS license expert but considering these issues putting TES under GPL by default seems the safest approach to be compliant, even if, with Apache/2-licensed PDFium being the default implementation, there is no actual integration with GPL libs by default.

Nevertheless, if you intend to fork this project and remove the GPL-related code in favor of PDFium only, feel free to put it under Apache-2 license.