TikaOCR

With TIKA-93 you can now use the awesome Tesseract OCR parser within Tika!

First some instructions on getting it installed. See Tesseract's readme.

Mac Installation Instructions

1. If you are lucky `brew install tesseract --with-all-languages --with-serial-num-pack` will work, if not, read on

Issues with Installing via Brew

If you have trouble installing via Brew, some options to try:

1. try typing `brew -v install tesseract --with-all-languages --with-serial-num-pack` 2. try to discern any make/configure errors. YMMV here. 3. if brew won't do it, you can also try and install Tesseract from source.

Tesseract won't work with TIFF files

If you are having trouble getting Tesseract to work with TIFF files, read this link. Summary:

1. uninstall tesseract `brew uninstall tesseract` 2. uninstall leptonica `brew uninstall leptonica` 3. install leptonica with tiff support `brew install leptonica --with-libtiff` 4. install tesseract `brew install tesseract --with-all-languages --with-serial-num-pack`

Installing Tesseract on RHEL

1. Add "epel" to your yum repositories if it isn't already installed
   1a. `wget https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm` (or appropriate version)
   1b. `rpm -Uvh epel-release-latest-7.noarch.rpm`
   2. `yum install tesseract` 3. To add language packs, see what's available `yum search tesseract` then, e.g. `yum install tesseract-langpack-ara`

Installing Tesseract on Ubuntu

1. `sudo apt-get update` 2. `sudo apt-get install tesseract-ocr` 3. To add language packs, see what's available then, e.g. `sudo apt-get install tesseract-ocr-fra`

Installing Tesseract on Windows

See UB-Mannheim.

Optimizing Tesseraact

There's some advice on the Tesseract github issues + wiki on ways to speed it up, eg #263 and #1171 and this wiki page.

Using Tika and Tesseract

Once you have Tesseract installed, you should test it to make sure it's working. A nice command line test:

```
tesseract -psm 3 /path/to/tiff/file.tiff out.txt
```

You should see the output of the text extraction in out.txt.

```
cat out.txt
```

Look for the text extracted by Tesseract.

Once you have confirmed Tesseract is working, then you can simply use the Tika-app, built with 1.7-SNAPSHOT or later to use Tika OCR. For example, try that same file above with Tika:

```
tika -t /path/to/tiff/file.tiff
```

That's it! You should see the text extracted by Tesseract and flowed through Tika.

Using Tika Server and Tesseract

Once you have Tesseract and a fresh build of Tika 1.7-SNAPSHOT (including Tika server), you can easily use Tika-Server with Tesseract. For example, to post a TIFF file to the server and get back its OCR extracted text, run the following commands:

in another window, start Tika server

```
java -jar /path/to/tika-server-1.7-SNAPSHOT.jar
```

in another window, issue a cURL request

```
curl -T /path/to/tiff/image.tiff http://localhost:9998/tika --header "Content-type: image/tiff"
```

Overriding the configured language as part of your request

Different requests may need processing using different language models. These can be specified for specific requests using the *X-Tika-OCRLanguage* custom header. An example of this is shown below:

```
curl -T /path/to/tiff/image.jpg http://localhost:9998/tika --header "X-Tika-OCRLanguage: eng"
```

Or for multiple languages:

```
curl -T /path/to/tiff/image.jpg http://localhost:9998/tika --header "X-Tika-OCRLanguage: eng+fra"
```

Overriding Default Configuration

When using the OCR Parser Tika will use the following default settings:

- Tesseract installation path = ""
- Language dictionary = "eng"
- Page Segmentation Mode = "1"
- Minmum file size = 0
- Maximum file size = 2147483647
- Timeout = 120

To changes these settings you can either modify the existing TesseractOCRConfig.properties file in tika-parser/src/main/resources/org/apache/tika/parser/ocr, or overriding it by creating your own and placing it in the package org/apache/tika/parser/ocr on your classpath.

It is worth noting that doing this when using one of the executable JARs, either the tika-app or tika-server JARs, will require you to execute them without using the *-jar* command. For example, something like the following for the tika-app or tika-server, respectively:

```
java -cp /path/to/your/classpath:/path/to/tika-app-X.X.jar org.apache.tika.cli.TikaCLI
```

```
java -cp /path/to/your/classpath:/path/to/tika-server-1.7-SNAPSHOT.jar org.apache.tika.server.TikaServerCli
```

OCR and PDFs

See also PDFParser notes for more details on options for performing OCR on PDFs.

Note: With Tika server, the PDFConfig is generated for each document, so any configurations that you may specify in the tika-config.xml file that you pass to the tika-server on startup are overwritten.

To go with option 1 for OCR'ing PDFs (run OCR against inline images), you need to specify configurations for the PDFParser like so:

```
curl -T testOCR.pdf http://localhost:9998/rmeta/text --header "X-Tika-PDFextractInlineImages: true"
```

To go with option 2 (render each page and then run OCR on that rendered image), you need to specify the ocr strategy:

```
curl -T testOCR.pdf http://localhost:9998/tika --header "X-Tika-PDFOcrStrategy: ocr_only"
```

Disable Tika OCR

Tika's OCR will trigger on images embedded within, say, office documents in addition to images you upload directly. Because OCR slows down Tika, you might want to disable it if you don't need the results. You can disable OCR by simply uninstalling tesseract, but if that's not an option, here is a tika.xml config file that disables OCR:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<properties>
  <parsers>
    <parser class="org.apache.tika.parser.DefaultParser">
      <parser-exclude class="org.apache.tika.parser.ocr.TesseractOCRParser"/>
    </parser>
  </parsers>
</properties>
```