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

```
brew install tesseract tesseract-lang
```

Issues with Installing via Brew

If you have trouble installing via Brew, you can try installing 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 tesseract-lang`

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 Tesseract

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

In Tika 2.x, users can modify configurations via a `tika-config.xml`. With the exceptions of the paths, we document the defaults in the following:

TesseractOCR Configuration

```
<properties>
  <parsers>
    <parser class="org.apache.tika.parser.DefaultParser">
      <!-- this is not formally necessary, but prevents loading of unnecessary parser -->
      <parser-exclude class="org.apache.tika.parser.ocr.TesseractOCRParser"/>
    </parser>
    <parser class="org.apache.tika.parser.ocr.TesseractOCRParser">
      <params>
        <!-- these are the defaults; you only need to specify the ones you want
             to modify -->
        <param name="applyRotation" type="bool">false</param>
        <param name="colorSpace" type="string">gray</param>
        <param name="density" type="int">300</param>
        <param name="depth" type="int">4</param>
        <param name="enableImagePreprocessing" type="bool">false</param>
        <param name="filter" type="string">triangle</param>
        <param name="imageMagickPath" type="string">/my/custom/imageMagicPath</param>
        <param name="language" type="string">eng</param>
        <param name="maxFileSizeToOcr" type="long">2147483647</param>
        <param name="minFileSizeToOcr" type="long">0</param>
        <param name="pageSegMode" type="string">1</param>
        <param name="pageSeparator" type="string"></param>
        <param name="preserveInterwordSpacing" type="bool">false</param>
        <param name="resize" type="int">200</param>
        <param name="skipOcr" type="bool">false</param>
        <param name="tessdataPath" type="string">/my/custom/data</param>
        <param name="tesseractPath" type="string">/my/custom/path</param>
        <param name="timeoutSeconds" type="int">120</param>
      </params>
    </parser>
  </parsers>
</properties>
```

OCR and PDFs

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

Note: With Tika server 1.x, 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. This behavior is changed in Tika 2.x, where the PDFConfig remembers settings from tika-config.xml and will only temporarily update custom configs sent via headers.

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"
```

Note: These two options are independent. If you set `extractInlineImages` to true and select an `OcrStrategy` that includes OCR on the rendered page, Tika will run OCR on the extracted inline images *and* the rendered page.

Disable OCR in Tika

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 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>
```

In Tika 2.x, you can selectively turn off OCR per parse programmatically by setting `skipOcr` on a `TesseractOCRConfig`. This will only affect that one call to parse.

```
        TesseractOCRConfig config = new TesseractOCRConfig();
        config.setSkipOcr(true);
        ParseContext context = new ParseContext();
        context.set(TesseractOCRConfig.class, config);

        Parser parser = new AutoDetectParser();
        parser.parse(inputStream, handler, metadata, context);
```

In Tika 2.x, with `tika-server`, add this header to skip OCR per request: `X-Tika-OCRskipOcr: true`

Optional Dependencies

Tika will run preprocessing of images (rotation detection and image normalizing with ImageMagick) before sending the image to tesseract if the user has included dependencies (listed below) and if the user opts to include these preprocessing steps.

To identify rotation

python must be installed with scikit-image and numpy

```
pip3 install numpy
```

```
pip3 install scikit-image
```

(As of January 5, 2021, there's a bug in the most recent numpy for Windows, specify 1.19.3: `pip3 install numpy==1.19.3`)

In Tika 2.0, `python3` must be installed and callable as `python3`.

Install ImageMagick

See: https://imagemagick.org/script/download.php

iOS: brew install imagemagick

Ubuntu: sudo apt install imagemagick

Windows: download the binary installer from the above page, e.g. https://imagemagick.org/download/binaries/ImageMagick-7.0.10-55-Q16-HDRI-x64-dll.exe

TODO: document how to configure these options in Tika