Transcript

This is a simple OCR application for scanned PDF's. It uses the Tesseract library's Java port for the OCR. PDFBox is used for splitting the PDF to pages and then save the pages as images and the result is saved as DOCX using docx4j.

Installation

Compilation

The project is Gradle-based, just issue

```
gradle build
```

to build it and/or:

```
gradle run
```

to run it.

A self-standing, so called "fat jar" can be created by issuing:

```
gradle fatJar
```

The JAR file will be created under `build/libs` and the name is `transcript.jar`.

The JAR contains some libraries (the Tesseract binding and the JavaFX libraries) that are not pure Java, but it contains the binaries for both Windows and Linux so the application can be used on these platforms. It possibly works on Mac as well, but this hasn't been tested.

The training data

There's one extra step that is needed: Tesseract needs training data. This is language-dependent information for the learning algorithm. The application needs the `<language code>.traineddata` files in the `tessdata` directory. They can be retrieved from github.com/tesseract-ocr/tessdata.

The following steps are needed:

1. change to the root directory of the project (`transcript`)
2. clone the repository above
    - it will create a directory called `tessdata` and place the contents of the repository there

OR

1. create a directory called `tessdata` in the project root (`transcript`)
2. download the contents of the repository above (or just the `tessdata` files of the languages that you would like to work with) and place the `traineddata` files for the required languages in `tessdata`

Usage

You can run the application by issuing:

```
java -jar <path-to-transcript>/transcript.jar
```

or you can simply double click on the JAR file if your system is configured that way. Please note, that the directory called `tessdata` with the training files need to reside in the working directory (where the application is started from), otherwise it won't find them and will be unable to run.

You need to browse for the PDF file. It will create a PNG image file for every page of the document and then a DOCX file with the same name as the original PDF file. The image files are kept but can be deleted when the transformation is complete.

When the image files have been created, a dialog window pops up that allows you to rotate pages of the document. Make sure that the text is upright on all pages, otherwise the OCR algorithm will not be able to recognize it.

If the PDF document doesn't require any page processing, then the checkbox of the image processing window can be unchecked. This will allow the entire process to progress without the need of closing the processing window between page extraction and the OCR process.