Steps for installing, training and testing Tesseract

Installing Tesseract

There are two ways to install Tesseract on OS X:

- **From source:**
    - Installing dependencies using homebrew:

```
brew install cairo pango
brew install leptonica
brew install icu4c
```

    - Installing Tesseract with training tools

```
git clone https://github.com/tesseract-ocr/tesseract
cd tesseract
./autogen.sh
./configure \
--with-extra-libraries=/usr/local/cellar/lib \
--with-extra-includes=/usr/local/cellar/include \
ldflags=-l/usr/local/cellar/lib \
cppflags=-l/usr/local/cellar/include

make
sudo make install

# not required if training tools are not to be
installed
make training
sudo make training-install
```

Alternatively, the dependencies may be installed using MacPorts as well. The ./configure command must be modified accordingly to point the libraries to /opt/local/lib and so on.

If there are errors while executing sudo make install stating that the build dependencies have not been built then the training tools will not be correctly installed.

- **Using Homebrew:**
    - Install Tesseract:
        ```
        brew install tesseract
        ```
    - Install Tesseract with training tools
        ```
        brew install --with-training-tools tesseract
        ```
      In your .bash_profile include :

```
export $TESSDATA_PREFIX=/usr/local/Cellar/tesseract/3.04.01_1/share
```

Training Tesseract

The first step is to create the training documents:
Copy this into a doc file in the font style Tesseract is to be trained for . It should preferably have a line spacing of 1.5 and 1pt spacing between the characters. Save this as a pdf file with the name as [lang].font-name.exp0.pdf, with lang being an ISO-639 three letter abbreviation for your language and then convert it to a 300dpi tiff file. (you can use imagemagick for this but sometimes the formatted document is not compatible.) The following is the command to convert the file using imagemagick:

```
convert -density 300 -depth 4 lang.font-name.exp0.pdf
                lang.font-name.exp0.tif
```

If you're adding multiple fonts, or bold, italic or underline, repeat this process multiple times, creating one doc → pdf → tiff per font variation.

The next step is to train tesseract using the generated documents:

- *Create box files using the above-generated tiff files through the following command:*

```
tesseract lang.font-name.exp0.tiff lang.font-name.exp0 batch.nochop makebox
```

This command generates a box file for the input tiff file. The box file is a UTF-8 encoded file containing the x,y coordinates of the boxes containing each letter. Each letter is placed on a separate line. Here, Tesseract would've recognised each character in the tiff file, some of them may be correct while the others may be wrong. Also, it maybe possible that some letters have been merged as one or completely overlooked in the output file. These must be rectified manually in the box file. Go through each letter one by one and identify the ones which have been incorrectly identified, merged or ignored by Tesseract and correct the

same. Also modify the x,y coordinates accordingly. Box-file editors may be used to modify them. Sample box-tif files.

- *Feed the box file to Tesseract:*

```
tesseract eng.font-name.exp0.tif eng.font-name.box nobatch box.train.stderr
```

- *Extract unicharset from the file:*

```
unicharset_extractor *.box
```

- *Create font_properties file:*

  It should list every font you're training, one per line, and identify whether it has the following characteristics: <fontname> <italic> <bold> <fixed> <serif> <fraktur>, such as:

```
eng.arial.box 0 0 0 0 0
```

- *Create shapetable:*

```
shapeclustering -F font_properties -U unicharset *.tr
```

- *Create training data:*

```
mftraining -F font_properties -U unicharset -O lang.unicharset *.tr
cntraining *.tr
```

- *Create required DAWG files:*

  Copy the word list and frequent words list into two separate files and then run the following commands:

```
wordlist2dawg word_list lang.word-dawg lang.unicharset

wordlist2dawg frequency_list lang.freq-dawg lang.unicharset
```

- *Combine the created files:*

  Rename normproto, Microfeat, inttemp, pffmtable files with language name prefix and run:

```
combine tessdata lang.
```

- *Moving the files to /usr/local/share:*

```
sudo mv eng.traineddata /usr/local/share/tessdata/
```

Testing Tesseract

To test Tesseract run the following command with the new language/font as argument: `tesseract image.tif output -l lang`