Quicksign OCRized Text Dataset (QS-OCR)

Figure 1: Quicksign Logo

- Quicksign OCRized Text Dataset (QS-OCR)
- Download
- Motivation
- Details
  - QS-OCR-Large
  - QS-OCR-Small
- Methodology
- How to reproduce
  - Using Docker
  - Manual installation

The Quicksign OCRized Text Dataset is a collection of more than 400,000 labeled text files that have been extracted from real documents using optical character recognition (OCR). Each file is associated to a class of interest such as email, advertisement or scientific publication.

It is based on the RVL-CDIP (Ryerson Vision Lab Complex Document Information Processing) dataset [1] and the Tobacco3482 dataset [2]. Both datasets are subsets of the large Truth Tobacco Industry Documents archive.

This README tries to follow the Datasheets for Datasets guidelines [3]. See the datasheet for more information.

First release: May 2019.

Download

Both datasets are available in the "releases" section.

[1] A. W. Harley, A. Ufkes, K. G. Derpanis, "Evaluation of Deep Convolutional Nets for Document Image Classification and Retrieval," in ICDAR, 2015.

[2] J. Kumar, P. Ye and D. Doermann, "Structural Similarity for Document Image Classification and Retrieval", in Pattern Recognition Letters, November 2013.

[3] T. Gebru, J. Morgenstern, B. Vecchione, J. W. Vaughan, H. Wallach, H. Daumé III and K. Crawford, "Datasheets for Datasets," in CoRR, 2018.

Motivation

At Quicksign, we thrive to deliver smart and powerful document analysis tools for digital onboarding. To do so, our R&D team applies state-of-the-art research from computer vision, deep learning and natural language processing. We work at lot with optical character recognition (OCR) and although we deploy lots of efforts to make it as accurate as possible, we sometimes have to deal with noisy text due to recognition errors. To our surprise, very few public datasets for text classification address this problem. From IMDB and Amazon reviews to Toxic Tweets classification, existing datasets deal with user-generated content which can be considered "clean".

Leveraging a previous dataset of more than 400,000 annotated document images, we applied Tesseract OCR to generate two new text datasets. We reuse the existing classification labels. By combining the generated text files and the existing labels, this repository constitutes a new text classification dataset. We hope this help the field go further into automated document image analysis.

Details

There are two versions of the dataset: QS-OCR-Large and QS-OCR-Small.

QS-OCR-Large

QS-OCR-Large is the result of running the Tesseract OCR tool on the documents from the RVL-CDIP dataset. It contains 400,000 labeled text documents in 15 classes:

- 0. Letter
- 1. Form
- 2. Email
- 3. Handwritten
- 4. Advertisement
- 5. Scientific report
- 6. Scientific publication
- 7. Specification
- 8. File folder
- 9. News article
- 10. Budget

- 11. Invoice
- 12. Presentation
- 13. Questionnaire
- 14. Resume
- 15. Memo

We use the same format as the original RVL-CDIP dataset and the same folder structure as to remain as compatible as possible. Train, validation and test splits are predefined and given in a text format:

```
path/to/the/text/file.txt category
```

QS-OCR-Small

QS-OCR-Small is the result of running the Tesseract OCR tool on the documents from the Tobacco3482 dataset. It contains 3,482 labeled text documents in 10 classes:

- 0. Advertisement (ADVE)
- 1. Email
- 2. Form
- 3. Letter
- 4. Memo
- 5. News
- 6. Note
- 7. Report
- 8. Resume
- 9. Scientific

We use the same folder structure as the original dataset to remain as compatible as possible. Samples for each class are stored in specifically named folder.

As there are no predefined train/val/test split, we encourage users to perform k-fold cross-validation on this dataset.

Be careful: there is some partial overlap between QS-OCR-Small and QS-OCR-Large (due to overlap between Tobacco3482 and RVL-CDIP). Take care to remove the common samples if you want to evaluate transfer learning performance.

Methodology

Thanks to the effort of the Truth Tobacco Industry Documents archive, document images from the datasets are well-oriented and relatively clean. It is quite straightforward to run the Tesseract OCR engine on such documents.

We used the version `4.0.0-beta.1` of Tesseract using the following parameters:

- LSTM engine for better accuracy (`--oem 1`),
- fully automatic page segmentation without orientation or script detection (`--psm 3`),
- English language (`-l eng`).

We relied on the Python library pytesseract for automation.

We did not preprocess further the original TIFF grayscale images since they were clean scanned documents to begin with.

The resulting text was not postprocessed. Although it could benefit from some level of spell checking, we chose to provide the true output of Tesseract OCR "as is".

How to reproduce

In addition to the generated text files, we also provide scripts to regenerate the text outputs from the document images based on the existing annotated RVL-CDIP and Tobacco3482 datasets. These scripts have been tested on Ubuntu 18.04 although they should also work on MacOS X and Windows (using WSL).

Using Docker

The simplest way to reproduce the dataset is using Docker.

After cloning the repository, run:

```
docker build . -t qs/ocrized
mkdir -p datasets && docker run --user $(id -u):$(id -g) -it --rm \
                       -v `pwd`/datasets:/work/datasets/ qs/ocrized
# --user $(id -u):$(id -g) sets the user in the container as the user
#             that runs the command (useful with the shared directory)
# -it activates interactive mode
# --rm deletes the container after usage
# -v mounts a volume that will be seen inside the container
```

Note: if you use Task, you can simply execute `task run`.

This will create a `datasets/` folder in the current directory and mount it in the container. Running the `tobacco3842.sh` or `rvl-cdip.sh` scripts will download and process the datasets as needed.

For example:

```
./tobacco3842.sh
# Downloading files from UMIACS server...
# Decompressing .zip archives...
#
# 2 archives were successfully processed.
# Generating filelist
# Do you want to run the OCR script now? [y/N] N
```

You can manually run the `to_text.py` script using Python if you want. See `python to_text.py -h` for usage information.

Manual installation

A manual installation needs a few requirements:

- tesseract, wget, zip, tar,
- a Python 3 interpreter,
- Python dependencies: joblib, tqdm, pytesseract, Pillow,
- *optionally:* the languages files for tesseract (e.g. `tesseract-ocr-fra`, `tesseract-ocr-ger` on Ubuntu).

Note: if you use Task, you can simply execute `task setup`.

Installation of Tesseract and other softwares is OS-dependent. On Ubuntu/Debian you can run the `setup.sh` script using root privileges or:

```
apt-get install -y wget unzip tesseract-ocr
```

Python dependencies can be installed using pip (`pip install -r requirements.txt`) or (**recommended**) using poetry:

```
# Use pip install poetry if you do not use poetry yet
poetry install
poetry shell
```

Running the `tobacco3842.sh` or `rvl-cdip.sh` scripts will download and process the datasets as needed.

```
./tobacco3842.sh
# Downloading files from UMIACS server...
# Decompressing .zip archives...
#
# 2 archives were successfully processed.
# Generating filelist
# Do you want to run the OCR script now? [y/N] N
```

You can manually run the `to_text.py` script using Python if you want. See `python to_text.py -h` for usage information.