Setting up a DjVu search engine for scanned documents with OCR A case study of Linde's dictionary

Janusz S. Bień

August 25, 2016

Abstract

It is now standard that scans published in digital libraries are accompanied by dirty OCR. It is also more and more common to provide a search engine for the OCRed texts. Some search engines highlight the hits on the scans, which is very useful. We intend to demonstrate that owing to the DjVu format and the Open Source search engine presented such functionality can be provided very easily and without additional costs.

1 Introduction

Although the DjVu format is renowned primarily for its high compression ratio, this is not its most important advantage in the present day, as this factor is no longer of crucial importance for desktop users, and even smartphones can handle large files without technical problems.

The unique feature of the DjVu format is the ability to address with a URL not only a document and not only a page (the latter is at least theoretically possible also with the PDF format), but also a page fragment in a specific view. This allows the users, especially the researches, to easily "quote" the relevant fragments of scanned words.

A search engine for DjVu documents has been developed under the supervision of the present author and has been in use since 2009 at `http://corpora.klf.uw.edu.pl/en/`; the usage statistics can be found at `http://www.klf.uw.edu.pl/` (the *Słowniki* section).

The system in question is based on the linguistically oriented Poliqarp system (Polyinterpretation Indexing Query and Retrieval Processor, `http://poliqarp.sourceforge.net/`). It has a client-server architecture, and its Web client is available in two variants: a text only one used for the Polish National Corpus (`https://bitbucket.org/jwilk/marasca`), and a graphical one, which uses the aforementioned feature of DjVu to highlight hits on the scans (`https://bitbucket.org/jwilk/marasca-wbl`); we call it simply Poliqarp for DjVu.

The Poliqarp query language has been inspired by the Corpus Query Processor, a component of the Corpus Workbench developed at the University of Stuttgart and allows for sophisticated use of regular expressions, cf. e.g. [3]. A power user can use them to circumvent the OCR errors or historical spelling differences, as illustrated e.g. in [2].

More background information can be found in [1]. Although the primary purpose of the system is to be run on a server, it can be also installed on a typical desktop using a GNU/Linux operating system. The technical documentation is available at `http://korpusy.klf.uw.edu.pl/doc/index.html`.

All the required software is provided on the basis of the GNU General Public License or another open source license.

2 Creating graphic page images

2.1 Scanning

scanhelper
`http://jwilk.net/software/scanhelper`
scan tailor
`http://scantailor.org/`
niektóre wakaty nadal zwichrowane!
OCR not yet

2.2 Scan adjustments

The frontmatter added by the reprint editor distorted the page numbers in the volumes with single pagination. It appeared also that some original fontmatter (bastard title[1]) has been skipped in the reprint. The reprint frontmatter has been moved to separate files, and the bastard title has been scanned from the PIW reprint (ref???).

Moreover the large 6th volume of the dictionary has been split into two smaller volumes. Again the reprint frontmatter has been moved to separate files, and the original frontmatter reconstructed.

The operation actually has been performed later on the DjVu files after the OCR stage using the DjVu Libre utilities (`http://djvu.sourceforge.net/`), namely djvm.

The results are available at `http://teksty.klf.uw.edu.pl/view/creators/Linde=3ASamuel_Bogumi==0142=3A=3A.html`.

3 Creating DjVu documents

The DjVu files serve several purposes:

[1] the bastard title, usually a single line in capital letters, precedes the full title, and takes a separate leaf with blank verso, `http://archive.org/details/practicetypogra11vinngoog` after `https://en.wikipedia.org/wiki/Half_title`

- They are used for OCR, cf. section 3.2 (page 3).
- If needed, there are used to create HTML with proper segmentation used to provide token coordinates for marasca.
- They are served to the users, cf. ???
- They are also used in the run time to create the graphic snippets, cf. ???. That is the primary reason why the files are needed on the Poliqarp server.

The DjVu output of didjvu can have bundled or unbundled form. It does it matter till the stage ???, the bundled form is slightly more convenient in the early stages.

For this stage we have to prepare two input files:

- metadata
- outlines

and of course the graphic files with the scans.

3.1 Converting to DjVu

didjvu
`http://jwilk.net/software/didjvu`

3.2 OCR

OCR was performed with Tesseract called by `ocrodjvu` in several passes with different language parameters. The empty pages were not subject to OCR.

Here is a sample call:

```
time
ocrodjvu -D -t chars
-e tesseract -l pol
-p 44 -j 4 --save-raw-ocr=hOCR6-1/hOCR6-1pol44 --save-script LindeIIGP6-1sauvola-clean_pol_4
```

where

- -D, –debug: To ease debugging, don't delete intermediate files (not really needed, used just in case)
- -t chars, –details=chars Record location of every line, every word and every character (actually -t words would be sufficient, used just in case)
- -e, –engine=engine-id Use this OCR engine.
- -p, –pages=page-range Specifies pages to process. page-range is a comma-separated list of sub-ranges. Each sub-range is either a single page (e.g. 17) or a contiguous range of pages (e.g. 37-42). Pages are numbered from 1.

- -l, –language=language-id Set recognition language. language-id is typically an ISO 639-2/T three-letter code.

  Tesseract >= 3.02 allows specifying multiple languages separated by "+" characters.

- -j, –jobs=n Start up to n OCR processes.

- –save-raw-ocr=output-directory Save raw OCR results (typically in the hOCR format) into output-directory. The directory must exist and be writable.

- –save-script=script-file Save a djvused script with OCR results into script-file.

The results was in the following form:

- a DjVu document with a hidden text layer

- hOCR files generated by Tesseract for individual pages

The resulting DjVu files are available at `http://teksty.klf.uw.edu.pl/20/`.

Unfortunately one cannot assume that the Tesseract output is fully correct. There was several cases where `&c` was represented literally.

At this stage it is already possible to create the character histogram with `https://bitbucket.org/jsbien/unihistext/`. The data for the histogram have to be exported from the resulting DjVu files with `djvutxt`.

3.3 Augmenting the DjVu files

There are three ways to augment the DjVu documents:

- outlines

- metadata

- thumbnails

The syntax of outlines is described in the man page of djvused.

Outlines may contain both local and external references allowing to switch easily from volume to volume, therefore the final URLs of the volume are to be decided at this stage.

Outlines originally has been prepared with djvusmooth and later updated by hand. The result was in the form of a djvused script.

djvusmooth can be also used for preparing the metadata.

Metadata are to be stored in the document, moreover for user convenience they will be provided also in a different format as corpus metadata, cf. stage ??? The keywords are not predefined, so the content can be easily adapted to the needs. It is quite natural to include the bibliographic description in the metadata.

Thumbnails

4 Serving the DjVu files

The DjVu files should be served unbundled, cf.

The client references the documents only by their numbers, which are converted to the appropiate URLs thanks to the <basename>.djvu.filenames file.

The files can be served before the corpus is created.

5 Creating poliqarp corpus auxiliary files

We need the following input files:

- tagset definition
- metadata definition
- actual metadata
- structure
- hOCR:
  - generated from the DjVu files, cf. `http://korpusy.klf.uw.edu.pl/doc/building.html`, e.g.:

    ```
    djvu2hocr --word-segmentation=uax29 v01/unbundled/index.djvu > v01/hocr/v01.hocr
    ```

  - generated directly by Tesseract
- addresses of the DjVu documents

The output consists of augmented hocr files.

The hOCR files are augmented with the structure and in the next stage only the modified version is used.

The corpus section are represented in hOCR as `class` parameter of `div` element, e.g.

```
<div class="ocrx_vol6">
 <div class="ocrx_vol6part1">
   <div class="ocrx_front">
    <div class="ocr_page" id="page_1"
         title="image &quot;/tmp/ocrodjvu.xmWOca/000000.tif&quot;; bbox 0 0 5306 6666; ppage
```

5.1 Preparing tagset for poliqarp

tagset depends on hOCR and XCES files used!

5.2 Preparing metadata for poliqarp

The keywords used in the metadata have to be specified in ????.

The actual metadata have the form of a directory tree

????

The names of metadata fields are localised later, cf.

5.3 Preparing document structure for poliqarp

`http://korpusy.klf.uw.edu.pl/doc/building.html`

```
annotate-hocr --in-place v01/hocr/v01.hocr v02/hocr/v02.hocr v03/hocr/v03.hocr < structure.t
```

dry-run, in-place

Another configuration file can define the documents structure, which can be then used in the queries to limit the search. The is the top level structure specified as page ranges, e.g. frontmatter, preface etc. The text units defined this way are called sections.

6 Creating poliqarp corpus XCES files

hOCR files are used indirectly, after converting to XCES.

The ??? is to be used for hOCR generated from the DjVu files and xhocr tools are to be used for the Tesseract hOCR files.

The results of converting the Tesseract files to XCES are available at `http://teksty.klf.uw.edu.pl/21/`. The files are in a sense incomplete, as they don't account for empty pages!

7 Creating poliqarp corpus

We need the following input files created in the previous step:

- tagset ???
- metadata
- actual metadata
- XCES files

veryfing
frequency list:
the script: marasca-wbl / misc / frequency-list

8 Creating marasca corpus

In this stage we need to provide the scans addresses and token coordinates to marasca.

We need the following input files created in the previous step:

- augmented (?) hOCR files
- addresses of the DjVu documents (created automatically?)

augment-djvu-corpus.py newcorpus v01/hocr/v01.hocr v02/hocr/v02.hocr v03/hocr/v03.hocr

```
f_filenames = try_create_file('djvu.filenames')
f_coordinates = try_create_file('djvu.coordinates')
f_pagesizes = try_create_file('djvu.pagesizes')
```

Information pages!!!

9 Running the poliqarp server

Configuration file

10 Running the marasca server

Configuring www server

11 Searched texts format

For the search engine we need both the page images in the DjVu format and the OCR results in some form convertible to XCES (XML Corpus Encoding Standard). As the pages of the documents will be accessed in random order, we need the so-called unbundled or indirect form of DjVu documents; this means that every page is stored in a separate file and can be served independently of others.

Actually two workflows are most practical:

- Scans and OCR to PDF, PDF to DjVu. When using an OCR program providing output in the PDF format with the text underlying the page images, Jakub Wilk's pdf2djvu program will convert them into DjVu documents with the hidden text layers. Later appropriate utilities will extract the required information.
- Scans to DjVu, OCR to DjVu and/or hOCR.

– Scans can be converted to DjVu using the original programs of the DjVuLibre bundle (c44 etc., cf. `http://djvu.sourceforge.net/`), or the more recent and more powerful didjvu program by Jakub Wilk, which in particular allows to select one of nine additional binarization methods provided by the Gamera library (cf. e.g. `http://gamera.sourceforge.net/doc/html/binarization.html`). At `http://teksty.klf.uw.edu.pl/6/` you can find for comparison the same scans processed with different binarization methods.

  It is worth noting the binary form of the so-called foreground (i.e. the proper text without illustration etc.) called stencil is inherent in every DjVu document and can be viewed by the user if needed.

– The DjVu documents obtained from the previous stage may be OCRed with Jakub Wilk's ocrodjvu program, which is now rather a misnomer: although originally intended to be used with the ocropus free OCR program (`https://github.com/tmbdev/ocropy`), now is primarily used with Tesseract (also a free program, `https://code.google.com/p/tesseract-ocr/`). The output of ocrodjvu may be just a DjVu document with the hidden text layer added, but it may consist also of the hOCR files produced by the OCR engines, which contain additional information not stored in a DjVu file; you can find some examples at `http://teksty.klf.uw.edu.pl/20/`. In consequence, when using hOCR we can produce more rich XCES files.

  This is the very workflow we intend to demonstrate at the conference, and we will apply it to Linde's dictionary scans available at `http://teksty.klf.uw.edu.pl/view/creators/Linde=3ASamuel_Bogumi==0142=3A=3A.html`.

12 Designing the tagset and describing the document structure

The OCR results in the hOCR forms can be interpreted as a sequence of words, i.e. text tokens consisting of character strings, with the following attributes:

- Language. For tesseract it is possible to specify several languages to be used for recognition. It this worth noting that German in contemporary typefaces is considered a different language than German printed in Fraktur; although not very logical, it is quite convenient in practice.
- Word recognition confidence, i.e. a numerical value between 0 and 1.
- Font properties.

To make these properties accessible from a query, it is necessary to create appropriate XCES tagset. For our demonstration we will used a tagset which is described by the following Poliqarp configuration file:

Linde.cfg

```
[attr]
# undefined, Polish, German, Russian:
lang = und pl de ru
# Latin normal, Latin Fraktur, Cyryllic:
script = latn latf cyrl
series = medium bold
shape = upright italic
# word confidence (representation proposed by Jakub Wilk):
wconf = 0 1 2 3 4 5 6 7 8 9

# typically used for the list of part of speech attributes:
[pos]
token = lang script series shape wconf
```

13 Hardware and software requirements

The system has no special hardware requirement, any contemporary server or even a desktop is sufficient.

The software requires GNU/Linux with a WWW server (tested only with Apache), the Django framework (https://www.djangoproject.com/) and Python.

For the demonstration we will use a virtual machine running Debian GNU/Linux. The machine will be later available for download as an OVA appliance.

References

[1] Bień, J.S.: Efficient search in hidden text of large DjVu documents. In: Advanced Language Technologies for Digital Libraries. Lecture Notes in Computer Science (Theoretical Computer Science and General Issues) (6699). Springer, pp. 1-14. ISBN 978-3-642-23159-9 http://bc.klf.uw.edu.pl/177/

[2] Bień, J, S.: The IMPACT project Polish Ground-Truth texts as a DjVu corpus. Cognitive Studies | Études Cognitives (14). pp. 75-84. ISSN 2080-7147. DOI: http://dx.doi.org/10.11649/cs.2014.008

[3] Przepiórkowski, A., Krynicki, Z., Dębowski, Ł., Woliński, M., Janus, D., Bański, P.: A Search Tool for Corpora with Positional Tagsets and Ambiguities. In the Proceedings of the Fourth International Conference on Language Resources and Evaluation, LREC 2004, pp. 1235–1238. http://nlp.ipipan.waw.pl/~adamp/Papers/2004-lrec/fcqp.pdf.