Donate to *Programming Historian* today!

Working with batches of PDF files

Moritz Mähr

Learn how to perform OCR and text extraction with free command line tools like Tesseract and Poppler and how to get an overview of large numbers of PDF documents using topic modeling.

Peer-reviewed CC-BY 4.0 Support PH

EDITED BY
Anna-Maria Sichani

REVIEWED BY
Catherine DeRose

Jack Pay

PUBLISHED | 2020-01-30
MODIFIED | 2025-10-14
DIFFICULTY | Medium
https://doi.org/10.46430/phen0088

Donate today!

Great Open Access tutorials cost money to produce. Join the growing number of people supporting *Programming Historian* so

we can continue to share knowledge free of charge.

A Table of Contents

Contents

- A Table of Contents
- Overview
  - Motivation
  - Scope
  - Objectives
- Prerequisites
  - Skills
  - Software
    - Windows 10
    - MacOS
    - Linux
    - Topic Modelling
  - Data
- Assessing Your PDF(s)
  - Text Recognition in PDF Files
  - Extract Embedded Text from PDFs
  - Extract Embedded Images from PDFs
  - Combine Images and PDFs into a Single PDF
  - Use Topic Modelling to Analyze the Corpus
    - Download the Corpus
    - Prepare and Clean Up the Corpus
    - Create the Topic Model
    - Evaluate the Topic Model
- Concluding Remarks
  - Alternatives

Overview

Motivation

Humanities scholars often work with text-based historical and contemporary sources. In most cases, the Portable Document Format (PDF) is used as an exchange format. This includes digital reproductions of physical sources such as books and photographs as well as digitally created documents. The digitisation of these objects increases their accessibility and availability. Archives have begun to digitise entire collections and make them accessible via the Internet. Even more dramatic is the increase in the amount of data in digitally created sources such as those necessary for corporate and government reporting. As a result, humanities scholars are increasingly exploring larger collections by means of Distant Reading and other algorithmic tools. However, PDF documents are only suitable for digital processing to a limited extent and must first be converted into plain text files.

Scope

If you meet one or more of the following criteria, this lesson will be instructive for you:

- You work with text-based sources and need to extract the content of the sources
- Your files are in PDF file format or can be converted to this file format
- You work with a large corpus and you do not want to touch each file individually (batch processing)
- You want to examine your corpus by the means of Distant Reading and therefore need it to be in plain text format
- You don't have access to commercial software, such as Adobe Acrobat Professional or Abbyy FineReader

Objectives

In more technical terms, in this lesson you will learn to:

- Recognize and extract texts in PDFs with Optical Character Recognition (OCR)
- Extract embedded texts from PDFs
- Extract embedded images from PDFs

- Combine images and PDFs into a single PDF file
- Do all of the above at once (batch processing) with a large corpus.
- Analyze a large corpus using Topic Modelling to get a quick overview of the topics it contains

> Tesseract OCR software used in this lesson supports over 110 languages including non-western languages and writing systems.

Prerequisites

Skills

You should feel comfortable using the command line of your computer. Windows users should take a look at Introduction to the Windows Command Line with PowerShell. MacOS and Linux users should take a look at Introduction to the Bash Command Line.

Software

Windows 10

Some components of the unix-based open source software used in this lesson do not run on Windows systems natively. Fortunately, since the Windows 10 Fall Creators Update there is a workaround. Open PowerShell as administrator and run `Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Windows-Subsystem-Linux`. Install Ubuntu 18.04 LTS from the Microsoft Store. To initialize the Windows Subsystem for Linux (WSL) click on the Ubuntu tile in the Start Menu and create a user account.[1]

> Follow these instructions carefully and do not lose your credentials. You will need them as soon as you run programs as administrator.

Once the WSL is up and running, navigate to your working directory (for example Downloads). Invoke `bash` through PowerShell and install all requirements via the built-in package manager Aptitude.

```
```

```
bash
```

```
sudo apt install ocrmypdf tesseract-ocr-all poppler-utils image
```

MacOS

Installing all the requirements without a package manager is cumbersome. Therefore install the Command Line Tools for Xcode and Homebrew first. It offers an easy way to install all the tools and software needed for this lesson.

```
xcode-select --install
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.co
brew install ocrmypdf tesseract-lang poppler imagemagick
```

Linux

On Ubuntu 18.04 LTS and most Debian-based Linux distributions you can install all requirements via `aptitude`.

```
apt install ocrmypdf tesseract-ocr-all poppler-utils imagemagic
```

Even though all tools used in this lesson are shipped with Ubuntu, an update is recommended.

Topic Modelling

The Topic Modelling in the case study is performed with the DARIAH Topics Explorer. It is a very easy to use tool with a graphical user interface. You can download the open source program for Windows, Mac and Linux here.

If you are using a Mac and receive an error message that the file is from an "unidentified developer," you can overwrite it by holding control while double-clicking it. If that doesn't work, go to Systems Preferences, click on Security & Privacy, and then click Open Anyway.

Data

Throughout this lesson you will work with historical documents from the First International Conference of Labour Statisticians from 1923. The data of all past conferences is provided by the International Labour Organization (ILO) and is publicly available.

To make it easier for you to navigate through the file system and create folders, here are some basic commands of the Bash Command Line:

- Display the path of the current folder with `pwd`
- Display the contents of the current folder with `ls`
- Display only PDF files in the current folder with `ls *.pdf`
- Create a folder named proghist with `mkdir proghist`
- Change to this folder with `cd proghist`
- Open your current folder with a file browser `open .` (On Windows use `explorer.exe .`)
- Change to the parent folder with `cd ..`
- Change to your users home directory with `cd`
- Paste the code snippets into explainshell.com to see what the code actually does

Throughout the lesson I will assume that 'proghist' is your working directory.

Save all files below to your working directory:

- Classification of industries
- Statistics of wages and hours of labour
- Statistics of industrial accidents
- Report of the Conference
- International labour review

To illustrate image extraction and PDF merging you will include one more files to our corpus that is not directly related to the First International Conference of Labour Statisticians from 1923.

- Speeches made at the ceremony on 21 October 1923

For the Topic Modelling of the case study you will download more files later in the lesson.

> Always make a backup copy of your data before using the commands in this course. Text recognition and combining PDFs can change the original files.

Assessing Your PDF(s)

In order to make this lesson as realistic as possible, you will be guided by a concrete historical case study. The study draws on the extensive collection of the International Labour Organization (ILO), in particular the sources of the First International Conference of Labour Statisticians.

You are interested in what topics were discussed by the labour statisticians. For this purpose you will want to analyze all available documents of this conference using Topic Modelling. This assumes that all documents are available in plain text.

First you will get an overview of our corpus. Large databases can create a false impression of evidence. Therefore, the documents must be subjected to qualitative analysis. For this you will use scientific methods such as source criticism. All documents are written in English and are set in the same font. `ILO-SR_N1_engl.pdf`, `ILO-SR_N2_engl.pdf`, `ILO-SR_N3_engl.pdf` and `ILO-SR_N4_engl.pdf` are part of the same series. In addition, you note that the `ILO-SR_N2_engl.pdf` file does not contain any embedded text. You also note that `23B09_5_engl.pdf` contains images. One of these images contains text.[2]

1. You will recognize the text of `ILO-SR_N2_engl.pdf`
2. You will extract the text from all PDF files
3. You will extract images from `23B09_5_engl.pdf`
4. For illustrative purposes, you will combine different images and documents into a single PDF document. This can be helpful if the

scanning process involves individual image files that are to be combined into a single document

5. You will analyze a lot of plain text files using Topic Modelling

Text Recognition in PDF Files

For the text recognition, you will use OCRmyPDF. This software is based on the state-of-the-art open source text recognition software Tesseract, which is maintained and further developed by Google. The software automatically recognizes the page orientation, corrects skewed pages, cleans up image artifacts, and adds an OCR text layer to the PDF. Only the document language must be given as a parameter.

```
ocrmypdf --language eng --deskew --clean 'ILO-SR_N2_engl.pdf' '
```

```
➜ pdftest ocrmypdf --language eng --deskew --clean 'ILO-SR_N2_engl.pdf' 'ILO-SR_N2_engl.pdf'
   INFO -    5: [tesseract] Image too small to scale!! (2x36 vs min width of 3)
   INFO -    5: [tesseract] Line cannot be recognized!!
WARNING -   19: [tesseract] lots of diacritics - possibly poor OCR
WARNING -   23: [tesseract] lots of diacritics - possibly poor OCR
WARNING -   39: [tesseract] lots of diacritics - possibly poor OCR
WARNING -   51: [tesseract] lots of diacritics - possibly poor OCR
WARNING - Some input metadata could not be copied because it is not permitted in PDF/A. You may
wish to examine the output PDF's XMP metadata.
   INFO - Optimize ratio: 1.00 savings: 0.0%
   INFO - Output file is a PDF/A-2B (as expected)
```

Figure 1: The status messages of the software indicate recognition errors in the OCR process.

The status messages of the software indicate recognition errors during the OCR process (see Figure 1). If certain errors occur systematically, it may be worthwhile to write a correction script. See Cleaning OCR'd text with Regular Expressions.

> OCRmyPDF has many useful parameters to optimize your results. See the documentation. The output might look slightly different depending on the version used.

To process all PDF files in your working directory at once. OCRmyPDF automatically skips PDFs that already contain embedded text.

```
find . -name '*.pdf' -exec ocrmypdf --language eng --deskew --c
```

Extract Embedded Text from PDFs

To extract the embedded texts from the PDF files use Poppler). It is a very powerful command line tool for processing PDF files that is used by many other programs.

```
pdftotext 'ILO-SR_N1_engl.pdf' 'ILO-SR_N1_engl.txt'
```

To process all PDF files in your working directory at once. The status message `Syntax Warning: Invalid Font Weight` means that the file contains formatting that does not meet the standard specifications of PDF. You can safely ignore this message.

```
find . -name '*.pdf' -exec pdftotext '{}' '{}.txt' \;
```

Once you have extracted all the embedded text from the PDFs, you can easily browse the text files. You can use the Windows Explorer, MacOS Finder, or a command line program like `grep`. You can display all the mentions of the term "statistics".

```
grep 'statistic' . -R
```

`grep` is also able to process complicated search queries (so-called regular expressions). For example, you can also search for all files containing either "labour statistics" or "wage statistics".

```
grep -E 'labour statistics|wage statistics' . -R
```

Regular expressions also include numbers. This is particularly interesting for historians. This command displays all years in the twentieth century.

```
grep -E '19[0-9][0-9]' . -R
```

Once you have successfully extracted all text from the PDF files, they can be further analyzed using methods of Distant Reading such as Topic Modelling. You will apply such methods to the case study later in this lesson.

Extract Embedded Images from PDFs

PDF is a container file format and can contain multiple embedded images per page. You can also use Poppler to extract those images. The program allows us to select a target format for the extracted images. It is recommended to you use a lossless image format like PNG when working with the images.

```
pdfimages -png '23B09_5_engl.pdf' '23B09_5_engl'
```

> For digitally created documents, Poppler extracts all embedded image files. This often includes image files that are outside the visible area or overlaid by other objects.

To process all PDF files in your working directory at once.

```
find . -name '*.pdf' -exec pdfimages -png '{}' '{}' \;
```

Poppler can only extract illustrations if they are available as individual images in the PDF file. If you want to extract illustrations from a scanned page take a look at this lesson: Extracting Illustrated Pages from Digital Libraries with Python.

Combine Images and PDFs into a Single PDF

Although OCRmyPDF can process image files directly, there are cases where you first want to combine the images into a PDF document.

Because most image formats do not support multiple pages, each page of a document has to be saved as a single file. With the widespread command line image editing software ImageMagick you can achieve this very easily.

```
convert '23B09_5_engl-002.png' '23B09_5_engl-004.png' '23B09_5_
```

To combine all images into a PDF file at once use the wildcard operator `*.png`. This step could take a few minutes.

```
convert '*.png' 'all-images-combined.pdf'
```

If you want to combine different PDF files, you can rely on Poppler. Poppler does this job much faster than ImageMagick and preserves attributes of the original documents.

```
pdfunite 'ILO-SR_N1_engl.pdf' 'ILO-SR_N2_engl.pdf' 'ILO-SR_N3_e
```

Use Topic Modelling to Analyze the Corpus

Now that you have performed all the steps of the PDF processing on some examples, you can return to the historical question of the case study. Which topics were discussed by the labour statisticians at the international conferences of the ILO? In order to answer this question using Topic Modelling, the following steps are necessary:

1. Download the corpus
2. Prepare and clean up the corpus
3. Create the Topic Model
4. Evaluate the Topic Model

Both the download and the processing of the corpus are time consuming and resource intensive. At

doi.org/10.5281/zenodo.3582736 you can download the collection as a ZIP file and go directly to step 3.

Download the Corpus

To avoid confusion create a new folder with `mkdir` and open it with `cd`.

```
mkdir case_study
cd case_study
```

You can download the corpus from the ILO website. All English documents contain 'engl' in the title. It's over a gigabyte of data. Depending on your internet speed this may take a while.

To automate this step you can use the following command line commands. This will download all English documents (340 files) at once.

```
curl https://webapps.ilo.org/public/libdoc/ilo/ILO-SR/ |
grep -o 'ILO[^"]*engl[^"><\/]*' |
uniq |
sed 's,ILO,https://webapps.ilo.org/public/libdoc/ilo/ILO-SR/ILO
xargs -n 1 curl -O < list_of_files.txt
rm list_of_files.txt
```

Prepare and Clean Up the Corpus

Now you can batch process all downloaded PDF files. First, perform text recognition on all files that don't have embedded text. Then extract all embedded text from the files. Depending on the performance of your computer, this step may take several hours.

```
find . -name '*.pdf' -exec ocrmypdf --language eng --deskew --c
find . -name '*.pdf' -exec pdftotext '{}' '{}.txt' \;
```

Create the Topic Model

In order to create a Topic Model with the DARIAH Topics Explorer, you don't need to have any deeper mathematical knowledge about the Latent Dirichlet Allocation (LDA) that is used.[3] Nevertheless, it is worth clarifying some implicit assumptions of the model before you begin:

- A corpus consists of documents. Each document consists of words. Words are carriers of meaning. The order (sentences, sections, etc.) of the words is very important to understand its content. But it is not factored in for the purposes of this analysis. Only the frequency of words in a document or corpus (or more precisely the co-occurrence of words) is measured
- You determine how many topics are present in the corpus.
- Each word has a specific probability of belonging to a topic. The algorithm finds the corresponding probabilities of the individual words
- Words that occur very frequently do little to discriminate between the individual topics. They are often function words such as 'and', 'but' and so forth. Therefore, they should not be included in the analysis
- Topic modeling using LDA is non-deterministic. This means that a different result can be obtained for each run. Fortunately, the result usually converges towards a stable state. Run it several times and compare the results. You will quickly see if the topics remain stable

Now open the DARIAH Topics Explorer and follow the steps given in the software. Then:

1. Select all 340 text files for the analysis.
2. Remove the 150 most common words. Alternatively, you can also load the file with the English stop words contained in the example Corpus of the DARIAH Topics Explorer.
3. Choose 30 for the number of topics and 200 for the number of iterations. You should play with the number of topics and choose a value between 10 and 100. With the number of iterations you increase the accuracy to the price of the calculation duration.
4. Click on 'Train Model'. Depending on the speed of your computer, this process may take several minutes.

Evaluate the Topic Model

The DARIAH Topics Explorer has a graphical user interface that makes it very easy to explore and evaluate the Topic Model and its thirty topics. In this run, the second topic looks like this (see Figure 2).

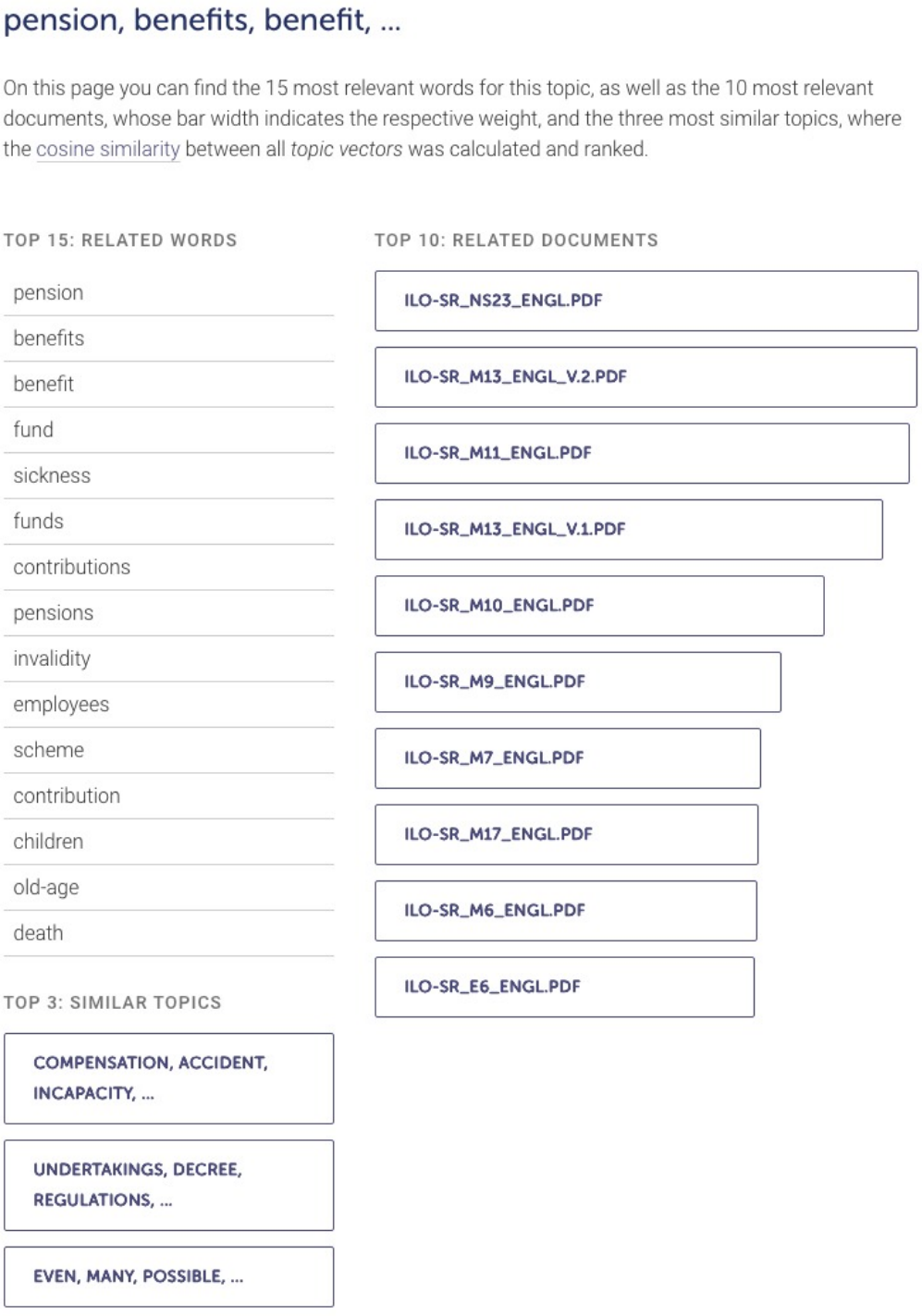

Figure 2: DARIAH Topics Explorer showing related words, related documents and similar topics of a single topic.

This topic deals with various social insurance schemes. Both old-age provision and unemployment benefits are included. The words are sorted in descending order of relevance and give a good overview of the topic. You can also see which documents have the highest

correspondence with this topic. As you can see when you look at related topics, this topic is close to accident insurance and legislation.

To further process or visualize the results with a spreadsheet program, click on the 'Export Data' button. The paper 'Parliament's Debates about Infrastructure' by Jo Guldi illustrates how Topic Modelling can be put to use for historical research.[4]

Concluding Remarks

Over the past decades, PDF has become the de facto standard for archiving and exchanging digital text documents.[5] However, this is not only the case for projects that focus primarily on digitized historical sources. For most digitally generated content, such as websites and interactive documents, as yet no generally accepted archiving formats have been established. Therefore, PDF is often used in these cases as well. Sometimes contemporary source documents present us with the same challenges as inferior scans of historical documents.

The Mueller Report is the official report documenting the findings and conclusions of the investigation by former Special Advisor Robert Mueller into Russian efforts to interfere in the 2016 presidential election in the United States, the allegation of conspiracy or coordination between Donald Trump's presidential campaign and Russia, and the allegation of obstruction of justice. As a technical analysis by Duff Johnson shows, the Mueller Report was digitally created, printed, scanned at least once, and sent for text recognition in an inferior version. Text and metadata were lost, which would have made working with the document much easier. It is unclear whether this deterioration in quality is intentional or due to complicated administrative procedures. In any case, it makes researchers' lives more difficult.[6]

Alternatives

This lesson focused on tools that are easy to use and are available as open source software free of charge. There are a lot of open source and commercial alternatives to process PDF files.[7] Getting Started with Topic Modeling and MALLET covers one of many alternatives for Topic Modelling.

1. If you run into troubles activating the WSL check out the troubleshooting, documentation, or the learning resources. ↩

2. In the case of a larger corpus, it is advisable to carry out random sampling instead of a detailed analysis. If no text is embedded in certain files, text recognition can be run over the entire corpus. Text recognition identifies embedded text when it is missing, and attempts to recognize it. ↩

3. If you still want to learn more, see Ganegedara, Thushan. "Intuitive Guide to Latent Dirichlet Allocation." *Medium*, August 23, 2018. https://towardsdatascience.com/light-on-math-machine-learning-intuitive-guide-to-latent-dirichlet-allocation-437c81220158. ↩

4. Guldi, Jo. "Parliament's Debates about Infrastructure: An Exercise in Using Dynamic Topic Models to Synthesize Historical Change." *Technology and Culture* 60, no. 1 (2019): 1–33. https://doi.org/10.1353/tech.2019.0000. ↩

5. The fourth chapter describes in detail the history of the PDF file format and the associated socio-technological upheaval, see Gitelman, Lisa. *Paper Knowledge: Toward a Media History of Documents.* Durham: Duke University Press, 2014. ↩

6. Johnson, Duff. "A Technical and Cultural Assessment of the Mueller Report PDF." *PDF Association* (blog), April 19, 2019. https://www.pdfa.org/a-technical-and-cultural-assessment-of-the-mueller-report-pdf/. ↩

7. Especially worth mentioning are the German wiki pages of the Ubuntu community about PDF and OCR. These pages contain references to free software for working with PDF files and improving text recognition. Unfortunately, there are no translations into other languages for these pages, so a translation service should be used. ↩

ABOUT THE AUTHOR

Moritz Mähr investigates the history of computers and migration at ETH Zurich in Switzerland.

SUGGESTED CITATION

Moritz Mähr, "Working with batches of PDF files," *Programming Historian* 9 (2020), https://doi.org/10.46430/phen0088.

Donate today!

Great Open Access tutorials cost money to produce. Join the growing number of people supporting *Programming Historian* so we can continue to share knowledge free of charge.

The *Programming Historian* (ISSN: 2397-2068) is released under a CC-BY license.

This project is administered by ProgHist Ltd, Charity Number 1195875 and Company Number 12192946.

ISSN 2397-2068 (English)

ISSN 2517-5769 (Spanish)

ISSN 2631-9462 (French)

ISSN 2753-9296 (Portuguese)