README – BWL Lernapp OCR- und Q&A-Pipeline

Dieses Dokument beschreibt die Skripte und deren Funktionen für die OCR-basierte Verarbeitung von PDF-Unterlagen (inkl. Bildinhalten), die Extraktion von Fragen und Antworten sowie den Export in formatierten Ausgaben für die BWL-Lernapp.

# Skript 1: install\_ocr\_text\_only.sh

(Shellscript zur Installation der benötigten Software und Einrichtung der Umgebung)

#!/usr/bin/env bash  
set -euo pipefail  
if ! command -v brew &>/dev/null; then  
 /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"  
fi  
  
brew install ocrmypdf tesseract tesseract-lang poppler  
  
python3 -m venv .venv\_ocr  
source .venv\_ocr/bin/activate  
  
pip install pymupdf pytesseract  
  
echo "Fertig. Aktiviere Umgebung: source .venv\_ocr/bin/activate"

# Skript 2: extract\_all\_qna\_inline.py

(Python-Skript zur Verarbeitung der PDFs: OCR, Frageextraktion, Generierung der Textdatei + Q&A JSON)

#!/usr/bin/env python3  
import re, tempfile  
from pathlib import Path  
import fitz, pytesseract  
  
IN\_DIR = Path("./pdf\_in")  
OUT\_TXT = Path("all\_text\_linear.txt")  
OUT\_JSON = Path("all\_qna.json")  
LANGS = "deu+eng"  
PSM = "6"  
  
# Erkennung von Fragen: Satz endet auf '?' oder enthält Rechen-Schlüsselwörter  
# Ablauf:  
# 1. PDF einlesen  
# 2. Text + Bildblöcke extrahieren (PyMuPDF)  
# 3. OCR auf Bildblöcken anwenden (pytesseract)  
# 4. Fragen und Antworten extrahieren  
# 5. Alles linear in TXT + Q&A JSON speichern

# Skript 3: pipeline\_qa\_text\_with\_image.py

(Pipeline für Q&A-Verarbeitung: Integration von Text- und Bildinhalten in eine Q&A-Logik)

#!/usr/bin/env python3  
import json, re  
  
# Lädt linearisierten OCR-Text  
# Extrahiert Fragen + Antworten  
# Stellt Q&A-Objekte mit Bildhinweisen her  
# Ergebnis: qanda\_with\_images.json

# Skript 4: export\_styled\_outputs.py

(Exportiert Ergebnisse in Markdown und Word mit farbigen Überschriften und formatierten Codeblöcken)

#!/usr/bin/env python3  
from docx import Document  
  
# Liest all\_text\_linear.txt und optional Q&A JSON  
# Teilt Inhalte in Kapitel zu je 10 PDFs  
# Exportiert Kapitel\_01.md, Kapitel\_02.md, ... und Kapitel\_01.docx etc.  
# Farbliche Styles: Heading1/2 via python-docx  
# Codeblöcke: Courier + grauer Hintergrund

# Nutzung

1. install\_ocr\_text\_only.sh ausführen (einmalig).  
2. PDFs in ./pdf\_in ablegen.  
3. extract\_all\_qna\_inline.py starten → erzeugt all\_text\_linear.txt + all\_qna.json.  
4. pipeline\_qa\_text\_with\_image.py optional zur Integration von Bildinformationen.  
5. export\_styled\_outputs.py ausführen → erzeugt formatierte Kapitel-Ausgaben.