



Mittelstand-Digital  
Zentrum  
Hannover

# Leitfaden

## Automatisierte Vertragsanalyse mit KI



Quelle: Thi/stock.adobe.com, generiert mit KI

Gefördert durch:



aufgrund eines Beschlusses  
des Deutschen Bundestages

Mittelstand-Digital

## Einleitung

Dieser Leitfaden bietet einen praxisnahen Einstieg für die Nutzung des Prototypen zur Vertragsanalyse des Mittelstand-Digital Zentrums Hannover. Hierbei sollen spezifische Daten aus den Verträgen ausgelesen werden und als sogenannte Metadaten abgespeichert werden. Der Leitfaden richtet sich an Anwendende mit Programmierkenntnissen und führt Schritt für Schritt von der Einrichtung bis zur fertigen Auswertung ihrer Dokumente.

**Zur Info:** Um die Metadatenextraktion möglichst robust abzubilden, wurden vier unterschiedliche Pipelines evaluiert. Hintergrund ist, dass die Ergebnisqualität stark von der Dokumentenqualität abhängt – insbesondere bei eingescannten Vorlagen und handschriftlichen Anteilen – und die Pipelines hier jeweils unterschiedliche Stärken und Schwächen zeigen.

Pipeline 1 nutzt **OpenAI Vision** über die API zur direkten Feldderkennung und -extraktion. In Pipeline 2 kommt **Mistral Annotation** – ebenfalls per API – zum Einsatz, um die Felder direkt zu extrahieren. Pipeline 3 verwendet **Mistral OCR** via API: Zunächst wird der Text per OCR ausgelesen, anschließend erfolgt die Feldderkennung durch ein Open-Source-Modell. Pipeline 4 verfolgt einen vollständig lokalen Ansatz: Der Text wird zunächst per **OCR** extrahiert und die Metadaten anschließend mithilfe eines lokalen **LLM** abgeleitet.



Quelle: Smile Studio AP/stock.adobe.com

## Hintergrund und Motivation

Viele Unternehmen verbringen viel Zeit damit, spezifische Daten (z. B. Laufzeiten, Kündigungsfristen, Partner) manuell in PDF-Verträgen zu suchen, um diese z. B. in Excel-Listen oder in ein CRM (Customer Relationship Management System (Software zur Kundendatenverwaltung)) zu übertragen. Dieses Programm übernimmt diese Aufgabe für Sie:

1. **Es liest Ihre PDF-Dateien ein (auch gescannte).**
2. **Es nutzt Künstliche Intelligenz (KI), um relevante Informationen zu verstehen.**
3. **Es vergleicht 4 KI-Ansätze (Pipelines), um das beste Ergebnis zu finden.**
4. **Es erstellt automatisch eine Excel-Tabelle mit allen Ergebnissen.**

## Vorbereitung (Was Sie brauchen)

Bevor Sie starten, stellen Sie sicher, dass folgende Dinge bereitstehen:

1. **Ein Computer** (Windows 11 empfohlen) mit Internetzugang.
2. **Die Programm-Dateien:** Der Ordner documents, in dem Sie Verträge ablegen.
3. **Ein API-Schlüssel:** Das Programm nutzt externe Cloud-KI-Dienste (OpenAI und Mistral).

Sie benötigen einen sogenannten **API-Key** (ein digitales Passwort), damit das Programm mit der KI kommunizieren kann. Den OpenAI-API-Key erhalten Sie unter <https://platform.openai.com/settings/organization/api-keys>. Für den Mistral-API-Key gehen Sie auf die Website <https://console.mistral.ai/home>. Auf beiden Plattformen müssen Sie sich anmelden und eine Kreditkarte hinterlegen. Anschließend können Sie einen API-Key generieren und diesen nutzen.

Die Kosten hängen von der Anzahl und Seitenzahl der Verträge ab. Für kleinere Mengen (z. B. 10–20 Verträge) bewegen sich die Kosten typischerweise im niedrigen einstelligen Euro-Bereich. Detaillierte Preisinformationen finden Sie auf den Plattformen von OpenAI und Mistral.

Um sicherzugehen, dass Sie nicht aus Versehen höhere Summen bei Ihrer Entwicklung bzw. Nutzung ausgeben, können Sie bei einigen Cloud-KI-Anbietern auch erstmal nur einen geringen Euro-Betrag als Credits erwerben („aufladen“).



Quelle: Sonia/stock.adobe.com, generiert mit KI

## Einmalige Einrichtung

### Schritt 1: Software installieren (Python & Git)

Damit Ihr Computer das Programm versteht und herunterladen kann, benötigen Sie zwei Programme:

#### 1. Python installieren

- Laden Sie Python herunter (<https://www.python.org/downloads/>).
- Starten Sie die Installation.
- **WICHTIG:** Setzen Sie im ersten Fenster unbedingt den Haken bei **Add Python to PATH**. Das ist entscheidend, damit die Befehle funktionieren.
- Klicken Sie auf **Install Now**

#### 2. Git installieren:

- Laden Sie Git herunter: <https://git-scm.com/download/win>
- Klicken Sie sich durch die Installation (die Standard-Einstellungen sind okay).

## Schritt 2: Das Programm herunterladen (Klonen)

Nun laden Sie den aktuellen Stand des Programms direkt von GitHub auf Ihren Computer.

1. Erstellen Sie einen neuen Ordner auf Ihrem Computer, wo das Programm liegen soll (z. B. c:\Projekte).
2. Öffnen Sie diesen Ordner.
3. Klicken Sie oben in die Adresszeile des Datei-Explorers, geben Sie `cmd` ein und drücken Sie Enter. Es öffnet sich ein schwarzes Fenster (das Terminal).
4. Geben Sie folgenden Befehl ein und drücken Sie Enter:  
`git clone https://github.com/iph-hannover/prowind.git vertragsanalyse`  
(Dieser Befehl lädt das Programm herunter und speichert es direkt in einem Ordner namens `vertragsanalyse`.)
5. Es wurde nun ein neuer Ordner namens `Vertragsanalyse` erstellt. Um in diesen neuen Ordner zu navigieren, geben Sie nun im Terminal ein:  
`cd vertragsanalyse`
6. Erstellen Sie jetzt hier einen Unterordner für Ihre Verträge. Geben Sie dafür im Terminal ein:  
`mkdir documents`

## Schritt 3: Den "digitalen Arbeitsplatz" einrichten

Wir befinden uns immer noch im Terminal-Fenster im Ordner `Vertragsanalyse`

1. Erstellen Sie den isolierten Programmbereich:  
`python -m venv venvllm`
2. Aktivieren Sie diesen Bereich  
`venvllm\scripts\activate`  
(Sie sehen nun eine `(venvllm)` am Anfang der Zeile.)
3. Installieren Sie die benötigten Hilfsprogramme:  
`pip install -r requirements.txt`  
(Dies kann einige Minuten dauern.)

## Schritt 4: Den API-Schlüssel hinterlegen

Um auf Cloud-basierte LLM wie Mistral AI oder OpenAI zuzugreifen, benötigt das Programm sogenannte API-Keys.

1. Erstellen Sie im Programm-Ordner (`vertragsanalyse`) eine neue Textdatei.
2. Nennen Sie diese Datei `.env`
3. Öffnen Sie die Datei mit dem Editor und fügen Sie Ihren Schlüssel ein, z. B.:  
`OPENAI_API_KEY = Ihr-Schlüssel`  
`MISTRAL_API_KEY = Ihr-Schlüssel`
4. Speichern und schließen

## Schritt 5: Ollama installieren (Optional für Pipeline 4)

Wenn Sie Pipeline 4 (lokale KI ohne Cloud) nutzen möchten, installieren Sie Ollama:

1. Ollama herunterladen und installieren:
  - Besuchen Sie <https://ollama.com/download>
  - Laden Sie die Windows-Version herunter und installieren Sie sie
2. KI-Modell herunterladen:
  - Öffnen Sie ein Terminal (`cmd`) und geben Sie ein:  
`ollama pull gemma3:12b`
3. Ollama-Server starten:
  - Ollama startet normalerweise automatisch im Hintergrund
  - Zum manuellen Starten:  
`ollama serve`

**Hinweis:** Wenn Ollama nicht installiert ist, wird Pipeline 4 übersprungen.

## Die tägliche Nutzung

Ist alles eingerichtet? Dann können Sie jetzt Ihre Verträge analysieren lassen.

### 1. Dokumente ablegen

Im Programm-Ordner finden Sie den erstellten Unterordner `documents`.

- Legen Sie alle PDF-Verträge, die Sie analysieren möchten, in diesen Ordner.
- Tipp: Löschen Sie alte Dateien aus diesem Ordner, die nicht erneut analysiert werden sollen.

### 2. Programm starten

1. Öffnen Sie das Terminal im Programm-Ordner (Adresszeile → `cmd` → Enter)
2. Starten Sie die Analyse mit einem Doppelklick auf die Datei:  
`run_all_pipelines.cmd`  
(Alternativ im Terminal: Tippen Sie `run_all_pipelines.cmd` und drücken Enter.)

Das Programm arbeitet nun automatisch und durchläuft alle **4 Analyse-Pipelines**:

- Pipeline 1: **OpenAI Vision** (nutzt `OPENAI_API_KEY`)
- Pipeline 2: **Mistral Annotation** (nutzt `MISTRAL_API_KEY`, optional)
- Pipeline 3: **Mistral OCR-VLM** (nutzt `MISTRAL_API_KEY`, optional)
- Pipeline 4: **RAG Doctling** (nutzt lokales KI-Modell via Ollama, optional)

Zum Schluss wird automatisch die Excel-Datei mit allen Ergebnissen erstellt.

### 3. Ergebnisse prüfen

Sobald das Programm fertig ist, finden Sie direkt im Hauptverzeichnis eine neue Excel-Datei `Alle_Pipelines_Standardisiert.xlsx`.

So lesen Sie die Excel-Tabelle:

- Zeile `_pipeline`: Zeigt an, welche KI-Methode (Pipeline 1-4) die Daten in dieser Zeile extrahiert hat
- Daten-Spalten: Hier stehen die extrahierten Informationen aus Ihren Verträgen:
  - **dokumentenname** - Name der PDF-Datei
  - **vertragsart** - Art des Vertrags (z.B. Kaufvertrag, Nutzungsvertrag)
  - **vertragspartner** - Namen der Vertragsparteien
  - **laufzeit** - Vertragslaufzeit
  - **kuendigungsfrist** - Kündigungsfrist
  - **zahlungsmodalitaet** - Zahlungsbedingungen
  - ... und weitere Felder

**Hinweis:** Welche Metadaten aus den Verträgen zu extrahieren sind, lässt sich im Code mit geringem Aufwand anpassen – insbesondere über die Metadatenbezeichnungen (Namen) sowie entsprechende Prompt-Anweisungen – und damit bei Bedarf auf spezifische Anforderungen zuschneiden.

### Qualitätskontrolle:

Da das Programm mehrere KI-Ansätze verwendet, können Sie die Ergebnisse vergleichen. Wenn verschiedene Pipelines unterschiedliche Werte für dasselbe Feld liefern, sollten Sie den Originalvertrag prüfen.

## Häufige Fragen (FAQ)

### Muss ich die Einrichtung jedes Mal machen?

Nein. Für die tägliche Nutzung reichen die Schritte unter „Die tägliche Nutzung“. Die Einrichtung (Schritte 1–4) machen Sie nur einmal.

### Das Terminal zeigt einen Fehler „OPENAI\_API\_KEY nicht gefunden“.

Prüfen Sie, ob die `.env`-Datei korrekt im Ordner **Vertragsanalyse** liegt und den API-Key enthält. Die Datei muss genau `.env` heißen (ohne `.txt` am Ende).

### Was passiert mit meinen Daten?

Die Daten werden zur Analyse an die KI-Anbieter (OpenAI, Mistral) gesendet. Stellen Sie sicher, dass dies mit den Datenschutzrichtlinien Ihres Unternehmens vereinbar ist. Pipeline 4 arbeitet komplett lokal ohne Datenübertragung.

### Kann ich nur einen Teil der Pipelines nutzen?

Ja. Wenn Sie nur den OpenAI-Schlüssel haben, läuft nur Pipeline 1. Die anderen Pipelines werden übersprungen. Das Programm funktioniert trotzdem und erstellt die Excel-Datei.

### Was macht Pipeline 4 (RAG Docting)?

Diese Pipeline nutzt ein lokales KI-Modell (kein API-Schlüssel nötig), benötigt aber die Installation von Ollama. Für den Einstieg können Sie diese Pipeline ignorieren.

### Die Ergebnisse sind nicht 100% korrekt.

Dies ist ein Prototyp zur Unterstützung. Die KI nimmt Ihnen die Suche ab, ersetzt aber (noch) nicht die fachliche Kontrolle. Nutzen Sie die Excel-Tabelle als Vorlage und prüfen Sie kritische Felder im Originalvertrag.



Quelle: Thapana\_Studio/stock.adobe.com

## Für Fortgeschrittene: Anpassung der Prompts

Wenn Sie die Extraktionsergebnisse verbessern möchten, können Sie die KI-Anweisungen (Prompts) anpassen.

### Wann sollte ich Prompts anpassen?

- Wenn ein bestimmtes Feld regelmäßig falsch extrahiert wird.
- Wenn Ihre Verträge eine spezielle Struktur haben.
- Wenn Sie zusätzliche Informationen extrahieren möchten.

### Wo finde ich die Prompts?

Die Prompts befinden sich in der Datei `src/field_configs.py`. Dort ist für jedes Datenfeld (z. B. **Vertragspartner**, **Laufzeit**) ein eigener Prompt definiert.

### Beispiel-Vertragspartner-Prompt:

```
"prompt": (  
    "Du erhältst einen Vertragstext. ALLE Verträge sind mit  
    der Mustermann GmbH abgeschlossen. "  
    "Finde die ANDERE Vertragspartei (nicht Mustermann).  
    Extrahiere NUR den Namen ohne Adresse. "  
    "WICHTIGE SUCHBEREICHE:\n"  
    "- Vertragsüberschrift: 'zwischen X und Y'\n"  
    "- Abschnitt 'Vertragspartner:' oder 'Grundstückseigentümer:'\n"  
    ...  
)
```

### Wie kann ich die Prompts anpassen?

1. Öffnen Sie die Datei `src/field_configs.py` mit einem Texteditor.
2. Suchen Sie das gewünschte Feld (z.B. `"name": "vertragspartner"`).
3. Ändern Sie den Text im `"prompt"`-Bereich. Achten Sie auf eine exakte Formulierung (siehe oberes Beispiel).
4. Speichern Sie die Datei.
5. Führen Sie die Analyse erneut aus.

**Hinweis:** Dies betrifft hauptsächlich Pipeline 4 (RAG Docling). Die anderen Pipelines verwenden eigene, fest kodierte Prompts in ihren jeweiligen Python-Dateien.