

# Drupal KI Lösungen: flexible Dokumentenzusammenfassung

## Inhaltsverzeichnis

### 1. Ausgangspunkt und Ziel des Projekts

1.1 Warum eine Dokumentzusammenfassung direkt in Drupal?

1.2 Beispiel-Projekt

### 2. Technische Grundlage

2.1 Verwendete Drupal-Module

2.2 OpenAI-Anbindung

### 3. Der Inhaltstyp

3.1 Zweck des Inhaltstyps

3.2 Benötigte Felder in der ersten Version

3.2.1 Dokument hochladen

3.2.2 Extrahierter Inhalt

3.2.3 Komprimierung

3.2.4 Sprachstil

3.2.5 Übersetzen nach

3.2.6 Zusammenfassung formatiert

### 4. Automator 1: Datei in Text umwandeln

4.1 Ziel des ersten Automators

4.2 Konfiguration des Datei-zu-Text-Automators

4.3 Besonderheiten der Textextraktion

## **5. Automator 2: Zusammenfassung erzeugen**

5.1 Ziel des zweiten Automators

5.2 Konfiguration des LLM-Text-Automators

5.3 Empfohlene Modellparameter

## **6. Der Prompt für den ersten Test**

6.1 Anforderungen an den Prompt

6.2 Prompt mit Drupal-Tokens und klaren Regeln

6.3 HTML-Regeln für die Ausgabe

## **7. Einen Inhalt für die Zusammenfassung anlegen**

## **8. Ausblick auf spätere Ausbaustufen**

8.1 Verarbeitung großer Dokumente mit Chunking

8.2 Weitere Eingabeformen

8.3 Weitere Ausgabeformate

8.4 Verbesserte Prompt-Kontrolle

## **9. Schätzung zu anfallenden Kosten im laufenden Betrieb für OpenAI**

Beispielrechnung mit GPT-4.1 mini

Annahme für die Rechnung

Rechnung

Spannbreite realistischer Kosten

## **10. Redaktionelle Qualitätssicherung**

## **11. Datenschutz**

# 1. Ausgangspunkt und Ziel des Projekts

## 1.1 Warum eine Dokumentenzusammenfassung direkt in Drupal?

Viele Unternehmen und Organisationen arbeiten regelmäßig mit längeren Dokumenten: Gutachten, Studien, Protokollen, Konzeptpapieren, Fachartikeln oder internen Arbeitsunterlagen. Häufig müssen diese Texte übersetzt, gekürzt und in eine gut lesbare Form gebracht werden.

Eine KI-gestützte Zusammenfassung direkt in Drupal kann hier sinnvoll sein. Der Vorteil liegt nicht darin, die redaktionelle Prüfung zu ersetzen, sondern einen wiederkehrenden Arbeitsschritt zu vereinfachen: Dokument hochladen, gewünschte Komprimierung, Stil und Sprache auswählen, Zusammenfassung erzeugen lassen und anschließend fachlich prüfen.

Statt jedes Dokument manuell in ein externes KI-Tool zu kopieren, entsteht ein eigener Drupal-Inhaltstyp für diesen Ablauf. Eingaben, Zwischenergebnisse und die erzeugte Zusammenfassung werden in Drupal-Feldern gespeichert und können später weiterverwendet, geprüft oder veröffentlicht werden.

## 1.2 Beispiel-Projekt

Für die erste funktionierende Version wurde der Ablauf bewusst einfach gehalten. Es geht zunächst nur um hochgeladene Dateien, später kann man auch ein Feld einfügen, das Links auf Dokumente im Netz entgegennimmt.

Der Inhaltstyp enthält folgende Felder: ein Dateifeld für das hochgeladene Dokument, Auswahlfelder für Komprimierung, Sprachstil und Zielsprache sowie ein Ziel-Feld für die formatierte Zusammenfassung. Zusätzlich gibt es im technischen Hintergrund ein Feld für den extrahierten Text, das den aus der Datei gewonnenen Inhalt aufnimmt.

Hier kann man für einfache Dokumente Plain Text einstellen, wir verwenden aber HTML, weil dadurch Strukturinformationen aus dem Dokument besser erhalten bleiben, wie zum Beispiel Überschriften, Absätze, Listen oder Hervorhebungen und sogar Tabellen.

Getestet wurde der Ablauf zunächst mit PDF- und ODT-Dateien. Zusätzlich können Word-Dokumente und reine Textdateien verarbeitet werden. Im Drupal Feld haben wir deshalb die Formate PDF, ODT, DOCX, DOC und TXT freigeschaltet. Hier kann man auch eine Größenbeschränkung in MB eintragen.

Allowed file extensions \*

Separate extensions with a comma or space. Each extension can contain alphanumeric characters, '.', and should start and end with an alphanumeric character.

File directory

Optional subdirectory within the upload destination where files will be stored. Do not include preceding or trailing slashes. This field supports tokens.

Maximum upload size

Enter a value like "512" (bytes), "80 KB" (kilobytes) or "50 MB" (megabytes) in order to restrict the allowed file size. If left empty the file sizes could be limited only by PHP's maximum post and file upload sizes (current limit **768 MB**).

Auch das Ausgabeformat wird zunächst nicht variabel gewählt. Die Zusammenfassung soll fest als HTML erzeugt werden. Das reduziert die Komplexität und erleichtert die Fehlersuche. Erst wenn dieser Grundablauf stabil funktioniert, können weitere Ausgabeformen ergänzt werden.

Der erste Zielprozess lautet also:

Datei hochladen → Dokumentinhalt strukturiert auslesen (im Hintergrund) → Einstellungen auswerten → HTML-Zusammenfassung erzeugen.

## 2. Technische Grundlage

### 2.1 Verwendete Drupal-Module

Die Umsetzung basiert auf vorhandenen Drupal-Modulen aus dem KI-Umfeld, wie AI, AI Automators, OpenAI Provider und AI File to Text.

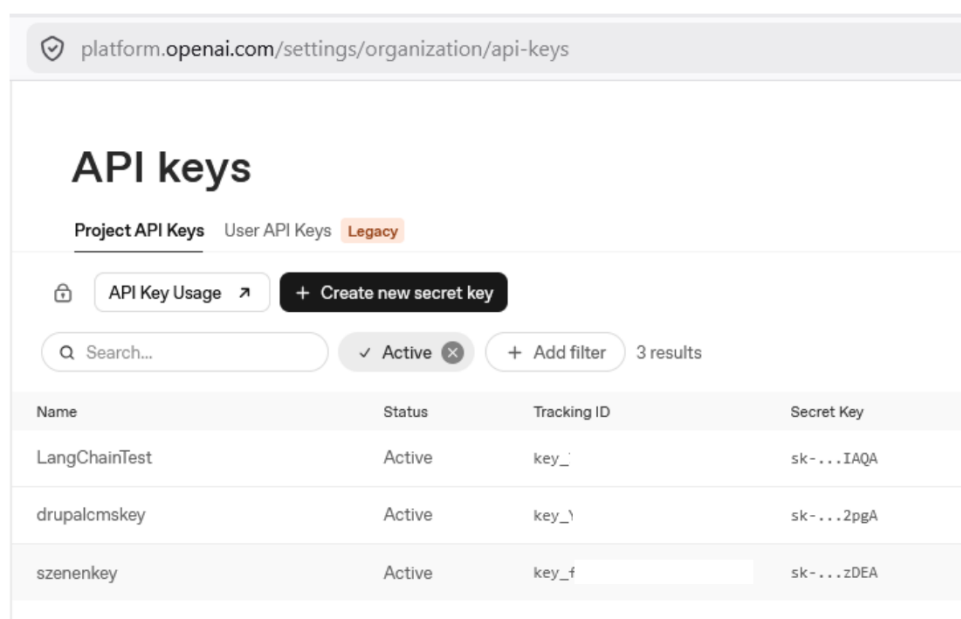
Der Vorteil: Es muss keine komplett eigene KI-Anwendung entwickelt werden. Stattdessen wird die KI-Funktion direkt in Drupal integriert und nutzt vorhandene Inhaltstypen, Felder und redaktionelle Abläufe.

Erste Grundlagen zu KI-Funktionen in Drupal wurden bereits in zwei früheren Beiträgen beschrieben:

- <https://www.montviso.de/blog/ai-content-suggestions-ki-gestuetzte-textoptimierung-direkt-drupal>
- <https://www.montviso.de/blog/kuenstliche-intelligenz-drupal-textgenerierung-aus-stichworten-mit-ai-automators-modul>

## 2.2 OpenAI-Anbindung

Wir binden für die Aufgabe OpenAI über eine API ein. Der API-Key wird nicht fest im Code hinterlegt, sondern über Drupal verwaltet. Dadurch bleibt die technische Konfiguration besser wartbar und sicherer.



Ein wichtiger Punkt ist der Unterschied zwischen ChatGPT und der OpenAI-API. Ein ChatGPT-Plus-Abo bedeutet nicht automatisch, dass auch API-Guthaben für technische Anbindungen vorhanden ist.

Für eine Drupal-Integration wird ein API-Guthaben benötigt. Ist dieses Guthaben aufgebraucht oder nicht eingerichtet, funktionieren API-Aufrufe aus Drupal nicht, auch wenn ChatGPT selbst weiterhin nutzbar ist.

Für Unternehmen ist das ein Vorteil: Die API-Nutzung kann getrennt vom ChatGPT-Abo geplant, begrenzt und überwacht werden. Dadurch lassen sich Kosten besser kontrollieren.

## **3. Der Inhaltstyp**

### **3.1 Zweck des Inhaltstyps**

Einer der größten Vorteile von Drupal sind Inhaltstypen. Mit diesen kann man festlegen, welche Felder und Bearbeitungsmöglichkeiten für eine bestimmte Art von Inhalt zur Verfügung stehen. Ein Blogbeitrag, eine Projektseite oder ein Veranstaltungseintrag ist jeweils ein eigener Inhaltstyp mit verschiedenen Feldern.

Für die KI-gestützte Dokumentenzusammenfassung wird in Drupal ebenfalls ein eigener Inhaltstyp angelegt.

Dieser Inhaltstyp bildet den gesamten Arbeitsablauf ab: Ein Dokument wird hochgeladen, die gewünschten Einstellungen werden ausgewählt und Drupal erzeugt daraus eine strukturierte Zusammenfassung.

Für Unternehmen ist das besonders interessant, weil solche Workflows nicht als externe Einzellösung neben der Website oder dem Intranet laufen müssen. Sie können direkt in bestehende Drupal-Strukturen eingebunden werden: mit Rollen, Rechten, redaktioneller Prüfung und gewohnten Bearbeitungsabläufen.

### **3.2 Benötigte Felder in der ersten Version**

Die erste Version enthält nur die Felder, die für den Grundprozess nötig sind: Datei hochladen, Verarbeitung steuern, Ergebnis speichern.

#### **3.2.1 Dokument hochladen**

Das Feld „Dokument hochladen“ nimmt die Datei entgegen, die zusammengefasst werden soll.

#### **3.2.2 Extrahierter Inhalt**

Aus der hochgeladenen Datei wird der enthaltene Dokumentinhalt als HTML gewonnen. Dieser Zwischenschritt ist wichtig, weil die KI-Zusammenfassung nicht direkt auf der Datei selbst basiert, sondern auf dem daraus extrahierten Inhalt.

Das Feld für den extrahierten Inhalt dient als technische Zwischenablage. Es macht nachvollziehbar, welcher Inhalt tatsächlich aus dem Dokument übernommen wurde. Das hilft bei der Kontrolle, besonders wenn Dateien unterschiedlich aufgebaut sind oder nicht sauber ausgelesen werden.

Im aktuellen Prototyp ist dieses Feld nicht für die Veröffentlichung gedacht. Es kann im Formular und im Frontend ausgeblendet werden, bleibt aber als technischer Zwischenschritt für den weiteren Workflow erhalten.

### **3.2.3 Komprimierung**

Mit dem Feld „Komprimierung“ wird festgelegt, wie stark der Ausgangstext gekürzt werden soll. Denkbar sind zum Beispiel Abstufungen wie ein Drittel, ein Viertel oder ein Sechstel des ursprünglichen Umfangs.

Ein kurzer Überblick für die Geschäftsführung benötigt eine andere Verdichtung als eine fachliche Arbeitszusammenfassung für ein Projektteam.

### **3.2.4 Sprachstil**

Das Feld „Sprachstil“ legt fest, in welcher Sprachebene die Zusammenfassung formuliert werden soll. Je nach Einsatzbereich kann die Ausgabe sachlich, einfach verständlich, lockerer oder für Experten ausfallen.

Dieselbe Information kann also für unterschiedliche Zielgruppen unterschiedlich aufbereitet werden, ohne dass der gesamte technische Ablauf verändert werden muss.

### **3.2.5 Übersetzen nach**

Mit dem Feld „Übersetzen nach“ kann die Zusammenfassung in eine andere Sprache übertragen werden.

Damit lässt sich der Workflow später gut für mehrsprachige Websites, internationale Projektteams oder interne Dokumentation erweitern.

### **3.2.6 Zusammenfassung formatiert**

Das Feld „Zusammenfassung formatierter Text“ nimmt das Ergebnis der KI auf. In der aktuellen Version wird die Ausgabe als HTML erzeugt und direkt in diesem Feld gespeichert.

## 4. Automator 1: Datei in Text umwandeln

AI Automators ist ein Drupal-Modul, mit dem sich Verarbeitungsschritte an Felder anhängen lassen. In diesem Zusammenhang ist ein Automator also ein konfigurierbarer Schritt: Wenn ein Inhalt gespeichert wird, kann das Modul ein Feld automatisch auswerten, Daten umwandeln oder ein anderes Feld mit einem Ergebnis befüllen.

### 4.1 Ziel des ersten Automators

Der erste Automator liest die hochgeladene Datei aus und schreibt den enthaltenen Dokumentinhalt als HTML in ein eigenes Drupal-Feld.

Das ist wichtig, weil die spätere KI-Zusammenfassung nicht direkt auf der Datei selbst basiert, sondern auf dem daraus gewonnenen strukturierten Inhalt.

### 4.2 Konfiguration des Datei-zu-Text-Automators

Für diesen Schritt wird ein Automator am Feld „Extrahierter Text“ eingerichtet. Als Quelle dient das Feld „Dokument hochladen“. Der Automator liest die hochgeladene Datei und schreibt den daraus gewonnenen Text in das Textfeld.

Der Ablauf ist:

Dokument hochladen → Datei auslesen → Dokumentinhalt als HTML in „Extrahierter Inhalt“ speichern.

In der Reihenfolge der Automatoren muss dieser Schritt vor der eigentlichen Zusammenfassung ausgeführt werden. Erst wenn der Text aus der Datei vorliegt, kann der zweite Automator daraus eine KI-Zusammenfassung erzeugen.

File to Text

Some field type might have many types to use, based on the modules you installed

^ AI Automator Settings

Automator Input Mode

Base Mode

For this Automator Type, only the Base Mode is available. It uses the base field to generate the content.

Automator Base Field

Dokument hochladen

This is the field that will be used as context field for generating data into this field.

Output Format

HTML

Plain text returns unformatted text. HTML includes tags (headings, lists, tables, bold, italic, etc.).  
Markdown returns Markdown-formatted output. JSON returns a structured JSON tree preserving docu...

## 4.3 Besonderheiten der Textextraktion

Die Textextraktion mit dem Drupal-Modul AI File to Text funktioniert auch ohne OpenAI-Anbindung. Die Datei wird zunächst über das entsprechende Drupal-Modul verarbeitet. Erst im zweiten Schritt wird ein KI-Modell benötigt.

Im Prototyp hat sich gezeigt, dass die Ausgabe als HTML sinnvoll ist, auch wenn dadurch mehr Tokens verbraucht werden und die Kosten bei OpenAI geringfügig steigen. Im Vergleich zu reinem unformatiertem Text bleiben dadurch einfache Strukturinformationen besser erhalten. Dazu gehören zum Beispiel Überschriften, Absätze, Listen oder Hervorhebungen und die Qualität der späteren Zusammenfassung ist besser.

Trotzdem bleibt eine Kontrolle wichtig. In der Entwicklungsphase ist es sinnvoll, das Feld mit dem extrahierten Inhalt zumindest im Testbetrieb nicht vollständig auszublenden, damit man Fehler in diesem Verarbeitungsschritt erkennt.

## 5. Automator 2: Zusammenfassung erzeugen

Der zweite Automator baut auf dem ersten auf.

## 5.1 Ziel des zweiten Automators

Ziel des zweiten Automators ist es, aus dem Feld „Extrahierter Text“ eine gut lesbare Zusammenfassung zu erzeugen. Dabei werden die zuvor gewählten Einstellungen berücksichtigt, zum Beispiel die gewünschte Komprimierung, der Sprachstil und gegebenenfalls die Zielsprache.

Das Ergebnis wird nicht nur als fließender Rohtext gespeichert, sondern als formatierter Text. In der ersten Version wird dafür HTML verwendet. Dadurch kann die Zusammenfassung bereits Überschriften, Absätze oder Listen enthalten und ist für die redaktionelle Weiterverarbeitung besser vorbereitet.

Der Automator ersetzt dabei nicht die fachliche Prüfung. Er erzeugt einen strukturierten Vorschlag, der anschließend kontrolliert, angepasst und bei Bedarf veröffentlicht werden kann.

## 5.2 Konfiguration des LLM-Text-Automators

Für die Zusammenfassung wird ein LLM-Text-Automator am Feld „Zusammenfassung formatiert“ eingerichtet. LLM steht für Large Language Model, also ein Sprachmodell, das Texte analysieren, kürzen, umformulieren und strukturieren kann. Dieser Schritt hat sowohl Konsequenzen auf die Qualität des Ergebnisses, als auch auf den Tokenverbrauch und damit die Kosten.

The screenshot shows the configuration interface for an LLM-Text-Automator. At the top, there is a toggle switch labeled "Enable AI Automator" which is turned on. Below it, a note says "If you want this value to be auto filled from AI". Then, a section titled "Choose AI Automator Type\*" contains a dropdown menu with "LLM: Text" selected. A note below this says "Some field type might have many types to use, based on the modules you installed". The main configuration area is titled "AI Automator Settings" and contains two sections. The first is "Automator Input Mode" with a dropdown menu showing "Advanced Mode (Token)". A note below this says "The Advanced Mode (Token) is available for this Automator Type to use multiple fields as input, you may also choose Base Mode to choose one base field." The second section is "Automator Prompt (Token)" which contains a text area with the following text: "Du bist ein präziser Assistent für Dokumentzusammenfassungen. Fasse den folgenden Text zusammen. Komprimierung: [node:field\_komprimierung]". A note at the bottom of this section says "The prompt to use to fill this field."

Als Grundlage verwendet der Automator den zuvor extrahierten Text im HTML-Format. Zusätzlich werden die Auswahlfelder des Inhaltstyps in den Prompt einbezogen. Dadurch kann dieselbe technische Struktur unterschiedliche Ergebnisse erzeugen: eine kurze Management-Zusammenfassung, eine einfachere Fassung oder eine stärker fachliche Variante.

## 5.3 Empfohlene Modellparameter

Für den Testbetrieb wird ein kostengünstiges Modell verwendet, das für viele redaktionelle Aufgaben ausreicht. Später kann je nach Dokumenttyp, Qualitätsanspruch und Kostenrahmen entschieden werden, ob ein leistungsstärkeres Modell sinnvoll ist.

Empfehlenswert sind niedrige Temperature-Werte, zum Beispiel 0,2 oder 0,3. Dadurch werden die Antworten weniger zufällig und bleiben näher am Ausgangstext. Auch die maximale Länge der Antwort sollte begrenzt werden, etwa über einen passenden Max Tokens Wert

The screenshot displays a configuration window for an AI provider. At the top, 'AI Provider' is set to 'OpenAI'. Below this, the 'Provider Configuration' section is expanded, showing the 'Model' set to 'gpt-4.1-mini'. Under the 'Settings' tab, three parameters are configured: 'Max Tokens' is 2048, 'Temperature' is 0.2, and 'Frequency Penalty' is 0. Each parameter has a text input field and a brief explanatory note below it.

| Parameter         | Value | Description                                                                                         |
|-------------------|-------|-----------------------------------------------------------------------------------------------------|
| Max Tokens        | 2048  | The maximum number of tokens that can be generated in the chat completion.                          |
| Temperature       | 0.2   | Sampling temperature 0-1. Higher values mean more random output.                                    |
| Frequency Penalty | 0     | Number between -2.0 and 2.0. Positive values penalize new tokens based on existing frequency in the |

## 6. Der Prompt für den ersten Test

Der Prompt ist die Arbeitsanweisung an das KI-Modell. Er legt fest, welche Aufgabe die KI erfüllen soll, welche Einstellungen berücksichtigt werden und welche Regeln für die Ausgabe gelten. Je klarer dieser Prompt formuliert ist, desto besser lässt sich das Ergebnis steuern.

### 6.1 Anforderungen an den Prompt

Für eine Dokumentenzusammenfassung ist wichtig, dass die KI nicht frei ergänzt oder interpretiert. Sie soll ausschließlich mit dem Text arbeiten, der aus dem hochgeladenen Dokument extrahiert wurde. Der Prompt muss deshalb klare Regeln nennen.

Damit der Prompt die gewählten Einstellungen aus dem Drupal-Inhalt für Komprimierung, Sprachstil und Zielsprache berücksichtigt, kommen s.g. Tokens ins Spiel. Das sind Platzhalter für Felder aus dem Inhaltstyp für diese Angaben. Auf diese Weise kann die Zusammenfassung später an den jeweiligen Zweck und an Zielgruppen angepasst werden kann. So ein Token lautet z.B. so `[node:field_sprachstil]` für das Feld Sprachstil.

### 6.2 Prompt mit Drupal-Tokens und klaren Regeln

Ein Prompt für die erste Version kann so aussehen:

Du bist ein präziser Assistent für Dokumentzusammenfassungen.  
Fasse den folgenden extrahierten Dokumentinhalt als HTML zusammen.

Komprimierung:  
`[node:field_komprimierung]`

Sprachstil:  
`[node:field_sprachstil]`

Übersetzen nach:  
`[node:field_übersetzen_nach]`

Regeln:

- Arbeite ausschließlich mit dem bereitgestellten Dokumentinhalt.
- Erfinde keine Informationen.
- Nutze die vorhandene HTML-Struktur als Orientierung.
- Erstelle eine eigenständige, gut gegliederte Zusammenfassung.
- Erhalte wichtige Namen, Orte, Daten, Zahlen, Argumente und Einschränkungen.
- Entferne Dopplungen.
- Wenn Informationen unklar oder widersprüchlich sind, benenne das.
- Schreibe keine Einleitung wie „Hier ist die Zusammenfassung“.
- Gib nur sauberes HTML zurück.

- Erlaubte HTML-Elemente: h2, h3, p, ul, ol, li, strong, em.
- Verwende keine Links, keine a-Tags, keine Scripts, keine iframes und keine

Extrahierter Dokumentinhalt im HTML-Format:  
[node:field\_extrahierter\_text]

Für Unternehmen ist daran vor allem wichtig: Die fachliche Steuerung steckt nicht nur im KI-Modell, sondern in der Kombination aus Drupal-Feldern und Prompt. Dadurch bleibt der Ablauf nachvollziehbar und kann später angepasst werden, ohne den gesamten Prozess neu zu entwickeln.

### 6.3 HTML-Regeln für die Ausgabe

In der ersten Version wird die Ausgabe als HTML erzeugt. Das ist sinnvoll, weil die Zusammenfassung dadurch mit Überschriften, Absätzen und Listen strukturiert gespeichert werden kann.

Gleichzeitig sollte die HTML-Ausgabe bewusst begrenzt werden. Die KI soll keine beliebigen HTML-Elemente erzeugen, sondern nur einfache, redaktionell gut kontrollierbare Strukturen. Erlaubt sind zum Beispiel Überschriften, Absätze, Listen und einfache Hervorhebungen. Aus Sicherheitsgründen verbieten wir Scripts, iframes, Inline-Styles oder automatisch erzeugte Links.

Die HTML-Regeln sind Teil der Qualitätssicherung. Sie sorgen dafür, dass das Ergebnis nicht nur inhaltlich, sondern auch strukturell in den redaktionellen Workflow passt.

## 7. Einen Inhalt für die Zusammenfassung anlegen

Ist der Inhaltstyp eingerichtet und sind die Automatoren konfiguriert, kann ein neuer Inhalt vom Typ „Dokumentenzusammenfassung“ angelegt werden.



# Leitfaden für erste Schritte mit der KNIME Analytics Plattform

## Inhaltsübersicht

- Was ist die KNIME Analytics Plattform?
- Einrichten der KNIME Analytics Plattform
- Erstellen des ersten Workflows
- Knoten und Arbeitsabläufe
- Weitere Beispiele
- Erweiterungen und Integrationen
- KNIME Forum

## Was ist die KNIME Analytics Plattform?

Die KNIME Analytics Plattform ist eine Open-Source-Software mit einer intuitiven, visuellen Oberfläche, die Analysen jeder Komplexitätsstufe ermöglicht – von der Automatisierung von Tabellenkalkulationen über ETL bis hin zum maschinellen Lernen.

## 8. Ausblick auf spätere Ausbaustufen

Die erste Version konzentriert sich bewusst auf einen einfachen und stabilen Ablauf: Datei hochladen, Text extrahieren, Zusammenfassung als HTML erzeugen. Damit entsteht eine belastbare Grundlage. Von dort aus kann der Workflow Schritt für Schritt erweitert werden.

### 8.1 Verarbeitung großer Dokumente mit Chunking

Je nach Modell gibt es technische Grenzen für die Textmenge, die in einem Aufruf verarbeitet werden kann. Deshalb kann man sehr große Dokumente nicht in einem Schritt verarbeiten, sondern man arbeitet mit Chunking. Dabei wird ein großes

Dokument in kleinere Abschnitte zerlegt. Diese Abschnitte werden zunächst einzeln zusammengefasst. Anschließend kann aus den Teilzusammenfassungen eine Gesamtsynthese erzeugt werden.

## 8.2 Weitere Eingabeformen

In der ersten Version werden nur hochgeladene Dateien verarbeitet. Denkbar wäre auch ein Feld für einen Link auf ein externes Dokument. Ebenso wäre eine direkte Texteingabe möglich, wenn kein eigenes Dokument hochgeladen werden soll. Der Inhaltstyp könnte dann eine Auswahl enthalten, ob die Quelle eine Datei, ein Link oder ein direkt eingefügter Text ist.

## 8.3 Weitere Ausgabeformate

Aktuell wird die Zusammenfassung als HTML erzeugt. Das ist für die redaktionelle Weiterverarbeitung in Drupal sinnvoll, weil Überschriften, Absätze und Listen direkt gespeichert werden können.

In späteren Versionen können weitere Ausgabeformate ergänzt werden. Denkbar sind zum Beispiel Exportfunktionen für LibreOffice Writer, Word-Dokumente, PDF oder tabellarische Formate. Je nach Einsatzzweck kann eine Zusammenfassung dann nicht nur im Drupal-System genutzt, sondern auch als Arbeitsdokument weitergegeben werden.

## 8.4 Verbesserte Prompt-Kontrolle

Ein weiterer Ausbauschritt betrifft die Kontrolle der KI-Anweisung selbst. Der Prompt kann sichtbar gemacht und dokumentiert werden, damit nachvollziehbar bleibt, mit welchen Vorgaben eine Zusammenfassung erzeugt wurde.

Das geht in Drupal mit einer einfachen `node_presave`-Funktion, welche die Platzhalter im Prompt durch die tatsächlich gewählten Werte für Komprimierung, Stil, Zielsprache ect. ersetzt und ihn so auf der Oberfläche ausgibt.

## 9. Schätzung zu anfallenden Kosten im laufenden Betrieb für OpenAI

Angenommen eine Firma lässt im Monat 100 PDF mit je durchschnittlich 10MB übersetzen und auf 1/3 zusammenfassen. Wieviele Token werden dafür benötigt und was kostet das nach momentanem Stand?

Man kann die Kosten nicht zuverlässig aus 10 MB PDF-Dateigröße berechnen. Für OpenAI zählt nicht die Dateigröße, sondern die Menge des extrahierten Textes, der an das Modell gesendet wird, plus die erzeugte Antwort. OpenAI erklärt Tokens als Verarbeitungseinheiten; grob gilt für englischen Text: 1 Token entspricht etwa 4 Zeichen. Für deutsche Texte kann die Tokenzahl je nach Wortlänge und Zusammensetzung etwas höher ausfallen.

## Beispielrechnung mit GPT-4.1 mini

Für den Prototyp verwenden wir **GPT-4.1 mini**. Laut OpenAI kostet dieses Modell derzeit:

- **Input:** 0,40 US-Dollar pro 1 Million Tokens
- **Output:** 1,60 US-Dollar pro 1 Million Tokens

## Annahme für die Rechnung

Für eine Zusammenfassung auf ein Drittel bedeutet das:

- pro PDF: 300.000 Input-Tokens
- erzeugte Zusammenfassung: 100.000 Output-Tokens
- pro Monat: 100 PDFs

Das wäre schon ein sehr umfangreiches Dokument. 300.000 Tokens entsprechen grob mehreren hunderttausend Zeichen extrahiertem Text.

## Rechnung

Pro Dokument:

- Input:  $300.000 \text{ Tokens} \times 0,40 \text{ US-Dollar} / 1.000.000 = \mathbf{0,12 \text{ US-Dollar}}$
- Output:  $100.000 \text{ Tokens} \times 1,60 \text{ US-Dollar} / 1.000.000 = \mathbf{0,16 \text{ US-Dollar}}$

Zusammen pro Dokument:

**0,28 US-Dollar**

Für 100 Dokumente pro Monat:

**28 US-Dollar pro Monat**

## Spannbreite realistischer Kosten

Eine PDF-Datei mit **10 MB** kann sehr unterschiedlich sein:

- viele Bilder, wenig Text → wenig Tokens
- viele Seiten Fließtext → viele Tokens
- gescannte Seiten ohne OCR → möglicherweise kaum auslesbarer Text
- PDF mit HTML-Struktur, Tabellen und vielen Formatierungen → mehr Tokens als reiner Text

Dadurch entsteht der Spielraum wie hier in der Tabelle:

| <b>Ausgelesener Text<br/>pro 10 MB Datei</b> | <b>Input</b>   | <b>Output</b>  | <b>Kosten für 100 PDFs</b> |
|----------------------------------------------|----------------|----------------|----------------------------|
| eher moderat                                 | 150.000 Tokens | 50.000 Tokens  | ca. <b>14 US-Dollar</b>    |
| umfangreich                                  | 300.000 Tokens | 100.000 Tokens | ca. <b>28 US-Dollar</b>    |
|                                              |                |                | ca. <b>75 US-Dollar</b>    |
| sehr groß                                    | 800.000 Tokens | 266.667 Tokens |                            |

Angenommen man benötigt ein qualitativ höheres Modell, wie GPT-5.4 mini oder GPT-5.5, dann kann der Preis schnell ein vielfaches betragen. Es muss also genau geprüft werden, ob sich die Kosten für Einrichtung und laufenden Betrieb rentieren. Spart man Lohnkosten ein, kann das schnell der Fall sein.

## 10. Redaktionelle Qualitätssicherung

Auch ein gut eingerichteter KI-Workflow ersetzt keine redaktionelle Prüfung. Die KI kann Arbeit beschleunigen, Texte strukturieren, übersetzen und Zusammenfassungen erzeugen. Ob das Ergebnis fachlich korrekt, vollständig ist und alle Fakten angemessen gewichtet sind, muss weiterhin ein Mensch prüfen.

Deshalb sollte der Workflow so gestaltet werden, dass KI-Ergebnisse nicht automatisch veröffentlicht werden. Besser ist ein redaktioneller Zwischenschritt: Zusammenfassung erzeugen, Ergebnis prüfen, bei Bedarf überarbeiten und erst dann freigeben.

Zur Qualitätssicherung gehört nicht nur die Prüfung der fertigen Zusammenfassung, sondern auch ein Blick auf den extrahierten Inhalt. Wenn bereits die Textextraktion Überschriften, Tabellen oder Absätze falsch übernimmt, kann auch die KI-Zusammenfassung nicht genau sein.

## 11. Datenschutz

Zusätzlich muss vor dem Einsatz geklärt werden, welche Dokumente überhaupt für eine Verarbeitung über eine externe KI-API geeignet sind. In diesem Workflow wird der aus der Datei extrahierte Text an OpenAI übergeben. Vertrauliche, personenbezogene oder rechtlich besonders sensible Inhalte sollten deshalb nicht unbedacht verarbeitet werden. Für Unternehmen gehört dazu eine klare Regelung, welche Dokumenttypen erlaubt sind, welche ausgeschlossen werden und welche Datenschutz- oder Vertraulichkeitsanforderungen gelten.

Gerade darin liegt der praktische Nutzen einer Drupal-Integration. Die KI-Funktion wird nicht isoliert eingesetzt, sondern in bestehende redaktionelle Abläufe eingebunden. So entsteht ein kontrollierbarer Prozess, der Arbeit erleichtert, ohne Verantwortung an die Technik abzugeben.