HTBL Krems

Höhere Lehranstalt für Informationstechnologie
Ausbildungsschwerpunkt: Systemtechnik

DIPLOMARBEIT

Anwendung für eine Firma

Ausgeführt im Schuljahr 2025/26 von:

Jacob Toifl	5AHIT-01
Michael Schaider	5AHIT-02

Betreuer:

Winkler Norbert, MSc
Winkler Norbert, MSc

Krems, am 01.04.2026

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Diplomarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche erkenntlich gemacht habe.

Krems, (Datum)

Verfasser/Verfasserinnen:

Jacob Toifl

Michael Schaider

DIPLOMARBEIT

Bestätigung der Abgabe

Abgabebestätigung

Datum	Name	Unterschrift

Genehmigung der Diplomarbeit

Approbation

Datum	Prüfer*in	Abteilungsleiter*in Direktor*in

DIPLOMARBEIT

Dokumentation

Verfasser*innen
Jacob Toifl, 5AHIT
Michael Schaider, 5AHIT
Abteilung
Informationstechnologie
Ausbildungsschwerpunkt: Systemtechnik
Schuljahr
2025/2026
Thema der Diplomarbeit
Anwendung für eine Firma
Kooperationspartner
MBIT Solutions GmbH

Aufgabenstellung

Realisierung

Ergebnisse

DIPLOMA THESIS

Documentation

Authors

Jacob Toifl, 5AHIT

Michael Schaider, 5AHIT

Department

Information Technology

Specialization: Systems Engineering

Academic year

2025/2026

Thesis Topic

Application for a Company

Cooperation Partner

MBIT Solutions GmbH

Task Description

Implementation

Results

Inhaltsverzeichnis

1. Präambel 8
1.1. Kurzfassung . . . 8
1.2. Abstract . . . 8
1.3. Team . . . 8
1.4. Danksagung . . . 8
1.5. Gendererklärung . . . 9

2. Einleitung 10
2.1. Ausgangslage . . . 10
2.1.1. Systemarchitekturen und Berechtigungssysteme . . . 10
2.1.2. KI-gestützte Klassifizierung und Suchlogiken . . . 10
2.2. Forschungsfrage . . . 10
2.2.1. Dynamische Zugriffskontrolle und Skalierbarkeit von Dokumentensystemen . . . 10
2.2.2. Automatisierte Dokumentenanalyse und Suchoptimierung . . . 10

3. Theoretische Grundlagen 11
3.1. Künstliche Intelligenz zur Dokumentenverarbeitung . . . 11
3.1.1. KI-gestützte Dokumentenklassifikation . . . 11
3.1.2. Architekturen zur Integration externer KI-Dienstleister . . . 13
3.1.3. Merkmalsextraktion und Embedding . . . 13
3.1.4. Modellfamilien zur Dokumenttypenerkennung . . . 17
3.1.5. Datenaufbereitung und Preprocessing für Dokumente . . . 19
3.2. Architekturen für Dokumentensysteme . . . 20
3.2.1. Suche und Filter-Architekturen . . . 20
3.2.2. Rollen- und Berechtigungssysteme . . . 22
3.2.3. Skalierbare plattformunabhängige Systemarchitekturen . . . 24
3.2.4. Sharepoint . . . 25
3.2.5. Microsoft Entra ID . . . 27

4. Dokumentation der Implementierung 31
4.1. Dokumentation - Grundlegend . . . 31
4.1.1. Test Umgebung . . . 31
4.1.2. Technologien . . . 31
4.2. Dokumentation - Funktionen . . . 31
4.2.1. Dokumenten-Upload . . . 31
4.2.2. Dokumenten-Klassifikation . . . 31
4.2.3. Dokumenten-Suche . . . 31
4.2.4. Benutzer- und Rollenverwaltung . . . 31
4.2.5. System-Logging und Monitoring . . . 31
4.2.6. API-Endpunkte . . . 31
4.2.7. Fehlerbehandlung und Ausnahmen . . . 31
4.2.8. Sicherheitsfunktionen . . . 31

5. Beurteilung 32
5.1. Bewertung der Implementierung . . . 32
5.2. Erfüllung der Forschungsfragen . . . 32
5.3. Kritische Reflexion und Grenzen . . . 32

6. Zusammenfassung und Ausblick 33
6.1. Zusammenfassung . . . 33
6.2. Ausblick . . . 33

I. Literaturverzeichnis 34

II. Abbildungsverzeichnis 36

III. Tabellenverzeichnis 37

IV. Quellcodeverzeichnis 38

A. Anhang 39
A.1. Arbeitsteilung . . . 39
A.2. Kapitelverzeichnis . . . 39
A.3. Projekttagebücher . . . 39
A.3.1. Projekttagebuch Max Mustermann . . . 39
A.3.2. Projekttagebuch Mex Musterjuan . . . 39
A.4. Besprechungsprotokolle . . . 40
A.5. Datenträgerbeschreibung . . . 42

1. Präambel

1.1. Kurzfassung

Die vorliegende Diplomarbeit beschäftigt sich mit der Entwicklung und Implementierung des Systems DropIT, einer modernen Lösung zur strukturierten, sicheren und effizienten Verwaltung von Dokumenten innerhalb einer Organisation. Ziel des Projekts ist es, eine nutzerfreundliche Anwendung zu schaffen, welche den Upload, die Klassifizierung, die Suche sowie die Organisation von Dokumenten zentralisiert und vereinfacht. DropIT integriert sich nahtlos in bestehende Microsoft-Dienste wie SharePoint und Entra ID, wodurch sowohl private als auch unternehmensinterne Anwender von einer verbesserten Übersichtlichkeit, Automatisierung und Datensicherheit profitieren.

1.2. Abstract

DropIT is a user-friendly document management system developed to make it easier to store, organize, and find digital files. The system works together with Microsoft services such as SharePoint and Entra ID, allowing secure login and central storage of documents. With AI-supported classification and metadata extraction, DropIT can automatically recognize document types and important information like contract periods or expiration dates. Features such as full-text search, filters, and a built-in reminder system help users manage documents faster and more efficiently. This thesis describes the idea, design, development, and evaluation of the system. The goal of DropIT is to offer a clear, reliable, and scalable solution that improves document handling for both private users and organizations.

1.3. Team

Das Projektteam besteht aus:

- **Jacob Toifl** – Projektleiter
- **Michael Schaider** – Projektmitglied

1.4. Danksagung

Wir möchten uns an dieser Stelle herzlich bei allen Personen bedanken, die uns während der Erstellung dieser Diplomarbeit unterstützt haben. Besonderer Dank gilt unserem Betreuer **Winkler Norbert, MSc**, für seine fachliche Beratung, seine Unterstützung im Entwicklungsprozess und seine wertvollen Rückmeldungen. Ebenso bedanken wir uns bei der Firma **MBIT Solutions GmbH** für die tolle Zusammenarbeit und die Bereitstellung der notwendigen Ressourcen und Infrastruktur, die maßgeblich zum Erfolg dieses Projekts beigetragen haben.

1.5. Gendererklärung

Zur besseren Lesbarkeit der Diplomarbeit wurde ausschließlich die männliche Form verwendet. Da Begriffe wie „Benutzerinnen und Benutzer“ den Text unleserlich machen, wurde es schlicht auf „Benutzer“ gekürzt, dies soll jedoch keine Geschlechterdiskriminierung zum Ausdruck bringen.

2. Einleitung

2.1. Ausgangslage

2.1.1. Systemarchitekturen und Berechtigungssysteme

Das ist die Ausgangslage von Jacob Toifl.

2.1.2. KI-gestützte Klassifizierung und Suchlogiken

Das ist die Ausgangslage von Michael Schaider.

2.2. Forschungsfrage

2.2.1. Dynamische Zugriffskontrolle und Skalierbarkeit von Dokumentensystemen

Das ist die Forschungsfrage von Jacob Toifl.

2.2.2. Automatisierte Dokumentenanalyse und Suchoptimierung

Das ist die Forschungsfrage von Michael Schaider.

3. Theoretische Grundlagen

3.1. Künstliche Intelligenz zur Dokumentenverarbeitung

3.1.1. KI-gestützte Dokumentenklassifikation

3.1.1.1. Definition von Dokumentenklassifikation

Bei der Dokumentenklassifizierung werden Dokumente bestimmten, zuvor definierten Klassen zugeordnet. Das Dokument wird zunächst erfasst, anschließend werden die enthaltenen Informationen ausgelesen und ausgewertet. So lässt sich erkennen, um welche Art von Dokument es sich handelt, wo es abgelegt werden soll, welche Daten daraus übernommen werden müssen und in welchen Workflow es anschließend einfließen kann. [1]

Zum Einsatz kommen dabei unter anderem OCR und KI, die selbst sehr feine Unterschiede zwischen verschiedenen Dokumentarten identifizieren können. Mithilfe von OCR werden Textinhalte aus Bilddateien ausgelesen, automatisch kategorisiert und in eine strukturierte Form gebracht. Dadurch können Dokumente und ihre Inhalte effizient gespeichert, verwaltet, durchsucht und ausgewertet werden. [1]

Die Begriffe Dokumentklassifizierung und Textklassifizierung werden häufig synonym verwendet, weisen jedoch einige Unterschiede auf, wie in Tabelle 3.1 ersichtlich.

Aspekt	**Textklassifizierung**	**Dokumentenklassifizierung**
Geltungsbereich	Analysiert nur Textinhalt.	Analysiert Text sowie Layout- und Bildelemente.
Data Input	Rein textliche Daten (Sätze, Absätze).	Gesamtes Dokument inkl. Bilder und Tabellen.
Anwendungsfälle	Sentiment, Themenzuordnung, Spam-Erkennung.	Rechnungen, Verträge, Formulare.
Techniken	NLP-Methoden.	Kombination aus NLP, Computer Vision und OCR.

Tabelle 3.1.: Textklassifizierung vs. Dokumentenklassifizierung [2]

Im Allgemeinen lässt sich sagen, dass Textklassifizierung eine Teilmenge der Dokumentenklassifizierung ist, die sich ausschließlich auf den Textinhalt konzentriert, während die Dokumentenklassifizierung einen umfassenderen Ansatz verfolgt. [2]

3.1.1.2. Funktion der Dokumentenklassifizierung

Die Dokumentenklassifizierung kann grundsätzlich auf zwei Wegen erfolgen: manuell oder automatisiert. Bei der manuellen Klassifizierung prüft eine Person die Dokumente, identifiziert inhaltliche Zusammenhänge und ordnet sie anschließend den entsprechenden Kategorien zu. Bei der automatischen Dokumentenklassifizierung kommen hingegen Verfahren des maschinellen Lernens bzw. Deep Learnings zum Einsatz. Ziel ist es, Dokumente ohne menschliches Eingreifen systematisch zuzuordnen. Für betriebswirtschaftliche Anwendungen ist es daher wichtig, die unterschiedlichen Dokumentarten sowie die damit verbundenen Geschäftsprozesse zu verstehen. [2]

Strukturierte Dokumente weisen klar definierte, einheitlich formatierte Daten auf (z. B. konsistente Nummerierung, Schriftarten und Layouts). Aufgrund dieser hohen Standardisierung lassen sich Klassifizierungsmodelle für solche Unterlagen vergleichsweise einfach entwickeln und die Ergebnisse sind gut prognostizierbar. [2]

Unstrukturierte Dokumente liegen in einem freien, wenig standardisierten Format vor. Beispiele sind Schreiben, Verträge oder Bestellungen mit variierendem Aufbau und sprachlicher Gestaltung. Durch diese Heterogenität ist die automatisierte Identifikation relevanter Informationen deutlich komplexer, was den Einsatz leistungsfähiger Klassifikationsverfahren erforderlich macht. [2]

3.1.1.3. Funktion der KI-basierten Dokumentenklassifizierung

Die automatisierte Klassifizierung von Dokumenten mit KI erfolgt typischerweise in mehreren aufeinanderfolgenden Schritten:

1. **Datensammlung und Beschriftung**
   Die Basis sind hochwertige, breit gefächerte Datenbestände. Dazu werden Dokumente aus unterschiedlichen Kategorien gesammelt und sauber mit passenden Labels versehen, damit Machine-Learning-Modelle sinnvoll trainiert werden können. [2]
2. **Vorverarbeitung und Feature-Erzeugung**
   Liegt ein Dokument als Scan oder Bild vor, wird der enthaltene Text zunächst per OCR (optische Zeichenerkennung) ausgelesen. Anschließend bereinigen NLP-Verfahren den Text, zerlegen ihn in Tokens und überführen ihn in aussagekräftige Merkmalsrepräsentationen. Parallel dazu wertet Computer Vision das Seitenlayout und visuelle Strukturen aus. [2]
3. **Training des Klassifikationsmodells**
   Überwachte Lernverfahren (etwa Transformer-Modelle oder CNNs) werden mit den gelabelten Beispielen trainiert, um wiederkehrende Muster zu entdecken. Das Modell lernt dabei, die gewonnenen Merkmale den jeweiligen Dokumentkategorien zuzuordnen. [2]

4. **Evaluation und Feintuning**
   Im Anschluss wird das Modell mit bislang unbekannten Testdaten geprüft, um Kennzahlen wie Genauigkeit, Präzision und Recall zu bestimmen. Durch Anpassung von Hyperparametern und ggf. Modellvarianten wird die Performance weiter verbessert. [2]
5. **Produktivbetrieb und laufende Anpassung**
   Nach der Implementierung ordnet das Modell neue Dokumente automatisch in Echtzeit den passenden Klassen zu. Über Nutzerfeedback und zusätzliche Trainingsdaten wird es regelmäßig nachtrainiert und kann seine Treffgenauigkeit im Zeitverlauf kontinuierlich steigern. [2]

3.1.2. Architekturen zur Integration externer KI-Dienstleister

3.1.3. Merkmalsextraktion und Embedding

3.1.3.1. Definition von Merkmalsextraktion

Die Merkmalsextraktion ist ein wichtiger Schritt in der Datenanalyse und im maschinellen Lernen. Dabei werden aus Rohdaten gezielt die Informationen herausgefiltert, die für eine spätere Auswertung oder ein Modell relevant sind. Ziel ist es, aussagekräftige Merkmale hervorzuheben und unwichtige oder störende Anteile zu reduzieren, sodass die Daten kompakter und besser nutzbar werden. [3]

Je nach Datentyp kommen dafür unterschiedliche Methoden zum Einsatz – von einfachen statistischen Verfahren bis hin zu maschinellen Lernverfahren, die Muster automatisch erkennen. Die gewonnenen Merkmale werden häufig in einer strukturierten Form zusammengefasst (z. B. als Merkmalsvektor) und bilden dann die Grundlage für weitere Schritte wie Klassifikation oder Vorhersagen. [3]

Wie in Tabelle 3.2 ersichtlich, werden im Folgenden wichtige Begriffe der Merkmalsextraktion mit kurzer Beschreibung dargestellt.

Begriff	**Beschreibung**
Merkmal	Eine quantitativ oder qualitativ erfassbare Eigenschaft, die ein Datenobjekt beschreibt.
Feature Extraction	Verfahren, bei dem aus Rohdaten gezielt die aussagekräftigen Eigenschaften herausgelöst bzw. abgeleitet werden.
Vektor	Geordnete Sammlung von Merkmalwerten, die ein Objekt in strukturierter Form repräsentiert.
Dimensionalität	Anzahl der enthaltenen Merkmale bzw. Einträge eines Vektors.

Tabelle 3.2.: Wichtige Begriffe der Merkmalsextraktion [3]

3.1.3.2. Funktion der KI-gestützten Merkmalsextraktion

KI-gestützte Merkmalsextraktion ist ein zentraler Bestandteil moderner Datenanalyse. Dabei werden automatisierte Verfahren eingesetzt, um aus großen Datensätzen gezielt relevante Informationen herauszuarbeiten. Durch den Einsatz unterschiedlicher KI-Algorithmen können Muster und Zusammenhänge präzise erkannt und die Daten effizient für weitere Analysen oder Modelle aufbereitet werden. [3]

Die in Tabelle 3.3 aufgeführten KI-Algorithmen sind zentrale Methoden für die Merkmalsextraktion.

Algorithmus	**Beschreibung**
Support Vector Machines (SVM)	Geeignet für Klassifikation und Regression; bestimmt eine optimale Trennlinie bzw. Entscheidungsgrenze zwischen Datenpunkten.
Decision Trees	Erstellen Entscheidungsbäume anhand von Merkmalen und Regeln, um Vorhersagen oder Klassen zu bestimmen.
Random Forest	Kombiniert viele Entscheidungsbäume und erhöht dadurch meist Genauigkeit und Stabilität der Ergebnisse.

Tabelle 3.3.: Relevante Algorithmen zur Merkmalsextraktion [3]

Maschinelles Lernen unterstützt die Merkmalsextraktion, indem Modelle aus Trainingsdaten lernen, welche Merkmale für eine Aufgabe wichtig sind, und weniger relevante Informationen ausblenden. Für Textdaten ist besonders Natural Language Processing (NLP) relevant, da damit Bedeutungen und Zusammenhänge in Sprache erkannt werden können. Wichtige Methoden sind Tokenisierung, Lemmatisierung und Named Entity Recognition (NER). Moderne Modelle wie BERT und GPT-3 nutzen große Datenmengen, um Muster im Kontext zu erfassen und Texte präziser zu analysieren. [3]

Text Mining ist ein Teilbereich der Merkmalsextraktion, der sich auf unstrukturierte Textdaten konzentriert. Ziel ist es, Texte computergestützt so auszuwerten, dass relevante Informationen herausgefiltert und unwichtige Inhalte reduziert werden, um daraus verwertbare Erkenntnisse zu gewinnen. Dabei werden insbesondere Verfahren des maschinellen Lernens und der Natural Language Processing (NLP) eingesetzt, um Muster, Bedeutungen und Zusammenhänge in großen Textmengen zu erkennen. Typische Anwendungsfelder sind unter anderem die Analyse von Kundenkommunikation sowie der Einsatz in Forschungskontexten, z. B. zur automatisierten Durchsicht umfangreicher Dokumente. [3]

3.1.3.3. Definition von Embedding

Ein Embedding ist eine Darstellung von Objekten wie Text, Bildern oder Audio als Zahlenvektoren in einem kontinuierlichen Vektorraum. Dabei sind ähnliche Inhalte im Raum näher beieinander angeordnet, sodass Machine-Learning-Modelle Ähnlichkeiten und Zusammenhänge leichter erkennen können. [4]

Embeddings werden in vielen Anwendungen genutzt, z. B. für Suche, Empfehlungssysteme, Chatbots oder Betrugserkennung. Im Unterschied zu handgebauten Merkmalen entstehen sie meist automatisch durch Lernverfahren (z. B. neuronale Netze), die Muster und Beziehungen direkt aus den Daten ableiten. Dadurch können Modelle nicht nur einzelne Wörter oder Elemente isoliert betrachten, sondern auch deren Kontext und Bedeutung besser erfassen. [4]

3.1.3.4. Prinzip und Funktionsweise von Embeddings

Bei der Embedding-Funktionsweise werden Rohdaten zunächst in ein numerisches Format überführt, weil viele ML-Algorithmen nur mit Zahlen arbeiten (z. B. Text als Bag-of-Words, Bilder als Pixelwerte oder Graphdaten als Matrix). Ein Embedding-Modell erzeugt daraus Vektoren – also Zahlenlisten –, die ein Objekt als Punkt in einem hochdimensionalen Raum repräsentieren. Jede Zahl steht für die Position entlang einer Dimension; je nach Aufgabe können es sehr viele Dimensionen sein. Ähnliche Objekte liegen im Vektorraum näher beieinander, und diese Nähe wird mit Ähnlichkeitsmaßen wie Cosinus-Ähnlichkeit oder euklidischer Distanz bewertet. [4]

In Abbildung 3.1 ist dieses Prinzip schematisch dargestellt: semantisch ähnliche Begriffe liegen im Vektorraum näher beieinander (z. B. Cat, Dog, Wolf), während thematisch andere Begriffe weiter entfernt positioniert sind (z. B. Apple, Banana). Die räumliche Distanz dient damit als Maß für Ähnlichkeit.

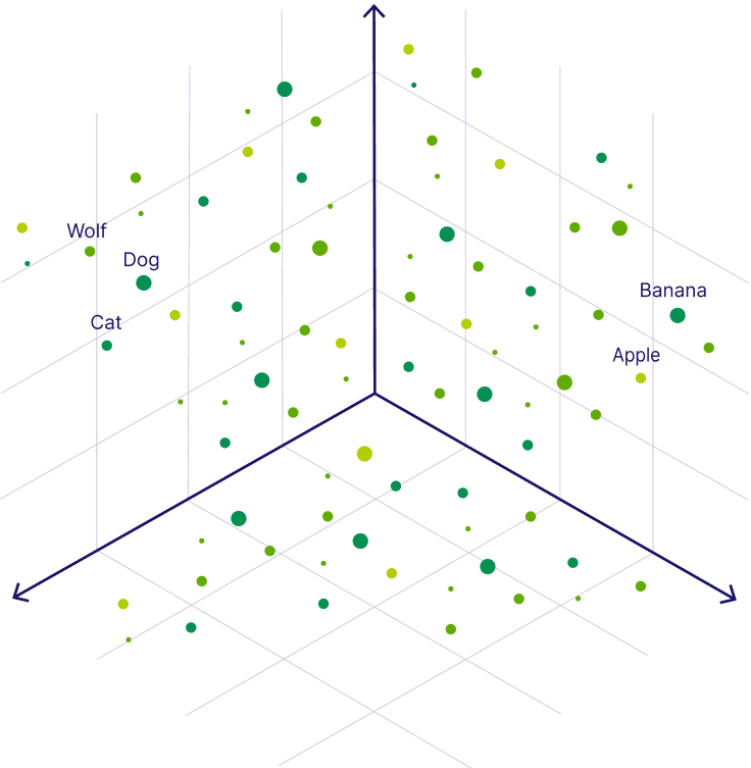

Abbildung 3.1.: Übersicht der Vektordarstellung von Embeddings [5]

Ein anschauliches Beispiel sind Wort-Embeddings: Wörter werden als Vektoren dargestellt, etwa „Papa“ und „Mama“. Auch wenn beide inhaltlich verwandt sind, wäre zu erwarten, dass „Vater“ im Vektorraum noch näher an „Papa“ liegt als „Mama“, weil die Bedeutung stärker übereinstimmt. [4]

Beispielhaft als Vektoren:

- „Papa“ = [0, 1548, 0, 4848, . . . , 1, 864]
- „Mama“ = [0, 8785, 0, 8974, . . . , 2, 794]

In Empfehlungssystemen werden sowohl Nutzer als auch Artikel als Embedding-Vektoren gelernt. Die Grundidee: Ein Nutzer und ein Artikel passen umso besser zusammen, je größer das Punktprodukt ihrer Vektoren ist. Der Empfehlungsscore wird dabei typischerweise so berechnet:

$$score = u * i$$

mit

- Empfehlungsscore $score$,
- Embedding des Nutzers u,
- Embedding des Artikels i,

Im Training werden diese Embeddings anhand historischer Interaktionen (z. B. Klicks, Käufe, Bewertungen) angepasst, sodass hohe Scores mit tatsächlichen Präferenzen möglichst gut übereinstimmen. Danach können Artikel mit den höchsten Scores als Top-N-Empfehlungen ausgegeben werden. [4]

3.1.3.5. Häufige Objekte für Embeddings

Embeddings sind flexible Repräsentationen, die sich auf unterschiedliche Datentypen anwenden lassen. Die häufigsten Objekte, die eingebettet werden können, sind in Tabelle 3.4 dargestellt.

Objekt	**Beschreibung**
Wörter	Dichte Vektorrepräsentationen einzelner Wörter, die Bedeutung und Kontextbeziehungen im Sprachkorpus abbilden.
Text	Vektoren für ganze Texte, die den Gesamtinhalt semantisch zusammenfassen und Vergleiche/Klassifikation erleichtern.
Bilder	Embeddings, die visuelle Merkmale und Bildinhalt als Vektor kodieren, z. B. für Ähnlichkeitssuche oder Objekterkennung.
Audio	Vektorrepräsentationen relevanter Klang- und Sprachmerkmale, nutzbar für Spracherkennung, Klassifikation oder Musikanalyse.
Graphen	Embeddings für Knoten oder ganze Netzwerke, die Struktur und Beziehungen erfassen, z. B. für Link Prediction oder Community Detection.

Tabelle 3.4.: Relevante Objekte für Embeddings [4]

3.1.3.6. Abgrenzung: Merkmalsextraktion vs. Embedding

Merkmalsextraktion beschreibt den allgemeinen Prozess, aus Rohdaten aussagekräftige Merkmale zu gewinnen oder abzuleiten, sodass die Daten für Analyse- oder Lernverfahren besser nutzbar sind. Dabei können unterschiedliche Methoden zum Einsatz kommen (z. B. statistische Kennwerte, regelbasierte Verfahren oder lernbasierte Ansätze), und das Ergebnis sind Merkmale, die eine Aufgabe gezielt unterstützen (z. B. Klassifikation oder Vorhersage).

Ein Embedding ist dagegen eine spezielle Form der Merkmalsrepräsentation: Objekte wie Wörter, Sätze, Bilder oder Nutzer/Artikel werden als dichte Vektoren in einem kontinuierlichen Vektorraum dargestellt. Die zentrale Idee ist, dass semantisch oder inhaltlich ähnliche Objekte im Vektorraum näher beieinanderliegen und dadurch über Ähnlichkeitsmaße vergleichbar werden.

3.1.4. Modellfamilien zur Dokumenttypenerkennung

Die Dokumenttypenerkennung ist ein zentraler Baustein in der KI-gestützten Dokumentenverarbeitung. Ziel ist es, eingehende Dokumente – etwa Rechnungen, Angebote, Verträge oder Formulare – automatisch einem definierten Dokumenttyp zuzuordnen, um nachgelagerte Verarbeitungsschritte wie Routing, Extraktion oder Validierung zu steuern. Je nach Anwendungsfall stehen dafür unterschiedliche Modellansätze zur Verfügung, die sich vor allem darin unterscheiden, welche Informationsquellen eines Dokuments sie nutzen: reinen Textinhalt, visuelle Merkmale einer Seite, Layout- und Positionsinformationen oder Kombinationen daraus. In der Literatur und in praktischen Systemen haben sich daher mehrere „Modellfamilien“ etabliert.

3.1.4.1. Textbasierte Modellfamilie

Textbasierte Verfahren formulieren die Dokumenttypenerkennung als Textklassifikationsproblem: Ein Dokument wird zunächst in eine rein textuelle Darstellung überführt (z. B. durch direkt extrahierbaren PDF-Text oder durch OCR bei gescannten Dokumenten). Anschließend wird dieser Text einer vordefinierten Klasse zugeordnet. Methodisch entspricht dies dem allgemeinen Vorgehen der Textklassifikation, bei dem Textinhalte analysiert und mittels gelabelter Beispiele in Kategorien eingeteilt werden. [6]

Innerhalb dieser Modellfamilie werden sowohl klassische Feature-basierte Ansätze (z. B. Bag-of-Words/gewichtete Termrepräsentationen) als auch neuere neuronale Modelle eingesetzt. Moderne Systeme nutzen häufig kontextuelle Sprachmodelle (Transformer), um neben Schlüsselbegriffen auch semantische Zusammenhänge zu berücksichtigen. Die Leistungsfähigkeit textbasierter Ansätze hängt in der Dokumentenverarbeitung jedoch stark von der Qualität der Texterfassung ab: OCR-Fehler, fehlende Sonderzeichen oder fragmentierter Text können die Klassifikation erschweren. Außerdem werden Layout- und Strukturinformationen eines Dokuments (z. B. Positionen von Feldern oder tabellarische Anordnungen) in einem rein textuellen Ansatz nur indirekt oder gar nicht abgebildet. [6]

3.1.4.2. Bildbasierte Modellfamilie

Bildbasierte Verfahren behandeln Dokumenttypenerkennung als Bildklassifikation: Seiten werden als gerenderte PDF-Seiten oder Scans (Seitenbilder) verarbeitet und auf Basis visueller Merkmale klassifiziert. Solche Ansätze nutzen typischerweise Convolutional Neural Networks (CNNs), die darauf ausgelegt sind, Muster in visuellen Daten zu erkennen und sich besonders für Klassifikationsaufgaben in der Computer Vision eignen. [7]

Ein wichtiges Argument für bildbasierte Ansätze ist, dass Dokumenttypen oft durch wiederkehrende visuelle Strukturen unterscheidbar sind (z. B. Formularcharakter, Briefkopf, typische Anordnung von Blöcken). In der Forschung wird die Dokumentbildklassifikation häufig auf etablierten Datensätzen evaluiert; als verbreitetes Beispiel gilt RVL-CDIP mit 16 Dokumentklassen und insgesamt 400.000 Dokumentbildern. Neben CNNs kommen zunehmend Vision Transformer (ViT) zum Einsatz, die Bilder in Patches zerlegen und diese ähnlich wie Token-Sequenzen verarbeiten, um globale Zusammenhänge besser abzubilden. [8] [9]

3.1.4.3. Layout-aware / dokumentenspezifische Transformer

Layout-aware Modelle erweitern textbasierte Ansätze um räumliche Layoutinformationen. Statt Text nur als Sequenz zu betrachten, wird zusätzlich modelliert, wo ein Token auf der Seite steht (z. B. über Bounding Boxes). Damit lässt sich die Dokumenttypenerkennung als Aufgabe formulieren, bei der Inhalt und Struktur gemeinsam berücksichtigt werden (z. B. typische Positionen von Überschriften, Adressblöcken, Tabellenbereichen). [10]

Ein prominentes Beispiel für diese Modellfamilie ist die LayoutLM-Klasse: In der Implementierung (z. B. in verbreiteten Frameworks) wird explizit ein zusätzliches Eingabesignal "bbox" erwartet, das die Bounding Boxes der Tokens enthält und damit die 2D-Positionen abbildet. [10]

3.1.4.4. Multimodale Foundation-Model-Familie (VLM/LLM)

Multimodale Foundation-Model-Ansätze nutzen große Modelle, die visuelle und häufig auch sprachliche Informationen gemeinsam verarbeiten können. In der Dokumenttypenerkennung werden Dokumentseiten (als Bild bzw. gerendertes PDF) typischerweise zusammen mit einer natürlichsprachlichen Aufgabenbeschreibung (Prompt) an ein Vision-Language-Modell übergeben, das den Dokumenttyp direkt ausgibt. Solche Verfahren sind besonders dann attraktiv, wenn eine schnelle Erweiterung auf neue Klassen erforderlich ist oder wenn man Zero-/Few-shot-Strategien erproben möchte. [11]

Dass diese Vorgehensweise praktisch umgesetzt wird, zeigen Anbieterbeispiele: Ein öffentliches Azure-Sample demonstriert explizit die Klassifikation eines Dokuments mithilfe eines Vision-fähigen GPT-4o-Modells (Prompt + Dokumentbild), also genau das Grundprinzip dieser Modellfamilie. [11]

3.1.5. Datenaufbereitung und Preprocessing für Dokumente

3.2. Architekturen für Dokumentensysteme

3.2.1. Suche und Filter-Architekturen

3.2.1.1. Suchalgorithmen im Information Retrieval

In diesem Kapitel werden die technologischen Grundlagen moderner Suchmaschinen behandelt. Ziel des Information Retrieval (IR) ist es, aus einer großen Menge unstrukturierter Daten jene Informationen zu finden, die einer Benutzeranfrage (Query) entsprechen. Dabei haben sich im Laufe der Zeit verschiedene Ansätze entwickelt.

3.2.1.2. Lexikalische Suchverfahren (Keyword Search)

Die lexikalische Suche ist der klassische Ansatz im Information Retrieval. Sie basiert auf dem Abgleich von Zeichenketten zwischen der Suchanfrage und den Dokumenten.

TF-IDF und Inverted Index

Ein grundlegendes Verfahren ist **TF-IDF** (Term Frequency - Inverse Document Frequency). Dieser Algorithmus bewertet die Relevanz eines Wortes, indem er zählt, wie oft es in einem Dokument vorkommt, gewichtet dies jedoch mit der Seltenheit des Wortes im gesamten Datensatz [**?**].

Das Rückgrat dieser Suche bildet meist ein sogenannter *Inverted Index*, der ähnlich wie ein Stichwortverzeichnis in einem Buch funktioniert.

BM25

Der heute am häufigsten eingesetzte Algorithmus ist **BM25** (Best Matching 25). Er ist eine Weiterentwicklung von TF-IDF und löst dessen Schwächen, indem er unter anderem die Dokumentenlänge berücksichtigt und eine Sättigungsgrenze für sehr häufige Begriffe einführt. BM25 gilt als Industriestandard für die Stichwortsuche.

3.2.1.3. Semantische Suchverfahren (Vector Search)

Während lexikalische Verfahren nur exakte Wortübereinstimmungen finden, zielen semantische Verfahren darauf ab, die *Bedeutung* (Semantik) einer Anfrage zu verstehen.

Embeddings und Vektorräume

Texte werden hierbei mithilfe von Machine-Learning-Modellen (z. B. BERT) in hochdimensionale Vektoren, sogenannte **Embeddings**, umgewandelt. In diesem mathematischen Vektorraum liegen Wörter mit ähnlicher Bedeutung räumlich nah beieinander (z. B. liegen die Vektoren für "Auto" und "PKW" nah beieinander, obwohl die Wörter völlig unterschiedlich geschrieben werden).

Suche mittels k-NN und HNSW

Um in diesen Vektorräumen effizient zu suchen, kommen Algorithmen wie **HNSW** (Hierarchical Navigable Small World) zum Einsatz. Dieser Algorithmus ermöglicht eine extrem schnelle *Approximate Nearest Neighbor* (ANN) Suche, selbst bei Millionen von Datensätzen [**?**].

3.2.1.4. Hybride Suchansätze (Hybrid Search)

In der Praxis zeigt sich oft, dass weder die reine Keyword-Suche noch die reine Vektorsuche alle Anwendungsfälle perfekt abdeckt.

- Die **Keyword-Suche** ist überlegen bei exakten Treffern (z. B. Artikelnummern, Eigennamen).
- Die **Vektorsuche** ist überlegen bei kontextbezogenen Fragen oder wenn der Nutzer den genauen Fachbegriff nicht kennt.

Daher kombinieren moderne Systeme beide Ansätze ("Hybrid Search"). Die Ergebnisse beider Algorithmen werden oft durch ein sogenanntes *Re-Ranking* (z. B. mittels Reciprocal Rank Fusion, RRF) zu einer finalen Ergebnisliste zusammengeführt.

3.2.1.5. Vergleich der Verfahren

Die folgende Tabelle fasst die wesentlichen Unterschiede zwischen den lexikalischen und semantischen Ansätzen zusammen.

Kriterium	**Lexikalische Suche (z. B. BM25)**	**Semantische Suche (Vektor)**
Funktionsweise	Abgleich exakter Zeichenketten	Mathematische Nähe im Vektorraum (Bedeutung)
Stärken	Findet exakte Begriffe (IDs, Namen); keine Trainingsdaten nötig	Versteht Synonyme, Kontext und Tippfehler
Schwächen	Scheitert bei Synonymen ("Auto" vs. "PKW"); versteht keinen Kontext	Langsamer; kann bei sehr spezifischen Begriffen ungenau sein
Einsatzgebiet	E-Commerce (Produkt-IDs), Code-Suche	Chatbots, FAQ-Suche, Empfehlungssysteme

Tabelle 3.5.: Vergleich zwischen lexikalischer und semantischer Suche

3.2.2. Rollen- und Berechtigungssysteme

3.2.2.1. Aufbau einer Berechtigung

Eine Berechtigung setzt sich grundsätzlich aus zwei zentralen Komponenten zusammen, die gemeinsam definieren, welcher Zugriff in einem System erlaubt oder untersagt ist.

Komponente	**Beschreibung**
Zu schützende Ressource	Beschreibt das Objekt, auf das sich eine Berechtigung bezieht. Dabei kann zwischen funktionalen und inhaltsbezogenen Ressourcen unterschieden werden. Die Definition der Ressource kann auf unterschiedlichen Granularitätsebenen erfolgen, beispielsweise global für ein gesamtes System oder atomar für einzelne Systembestandteile.
Zugehörige Operation auf der Ressource	Legt fest, welche konkrete Handlung in Bezug auf die definierte Ressource durch eine Berechtigung erlaubt oder eingeschränkt wird. Die Operation bestimmt, welche Aktionen eine Identität auf der jeweiligen Ressource ausführen darf.

Tabelle 3.6.: Komponenten einer Berechtigung

[12]

3.2.2.2. Arten von Berechtigungen

Berechtigungskonzepte bilden eine wesentliche Grundlage für die Zugriffskontrolle in IT-Systemen. Die elementarste Form stellt dabei die *binäre Berechtigung* dar. Sie beschreibt eine eindeutig definierte Zugriffserlaubnis, die sich jeweils auf ein konkretes Objekt und eine einzelne Operation bezieht. Der Zugriff ist entweder erlaubt oder verweigert, wodurch diese Berechtigungsform ausschließlich zwei mögliche Zustände kennt. [12]

Aufgrund dieser klaren Struktur eignen sich binäre Berechtigungen besonders für die technische Überprüfung von Zugriffsrechten. Umfangreichere Berechtigungsmodelle, die mehrere Objekte oder Operationen umfassen, werden vor der eigentlichen Zugriffskontrolle in einzelne binäre Berechtigungen zerlegt. Die endgültige Zugriffsentscheidung erfolgt anschließend ausschließlich auf Basis dieser elementaren Berechtigungseinheiten. [12]

Dieses Vorgehen findet auch bei weiterführenden Steuerungsmechanismen Anwendung, beispielsweise bei der Verwendung von Rollen oder Profilen. Solche abstrakten Konzepte werden zunächst auf konkrete Berechtigungen abgebildet und anschließend in mehrere binäre Einzelrechte aufgelöst, die festlegen, ob eine bestimmte Handlung erlaubt oder untersagt ist. [12]

Neben binären Berechtigungen existieren sogenannte *Wertberechtigungen*, bei denen mehrere Einzelrechte in einem numerischen Wert zusammengefasst werden. Ein verbreitetes Anwendungsbeispiel hierfür ist das Berechtigungssystem von UNIX-Dateisystemen. Dort

repräsentiert ein Zahlenwert die Kombination einzelner Zugriffsrechte wie Lesen, Schreiben und Ausführen. Der Wert 7 steht für den vollständigen Zugriff und ergibt sich aus der Addition der einzelnen Rechtewerte. [12]

Wertberechtigungen stellen somit eine kompakte Darstellungsform mehrerer binärer Berechtigungen dar. Obwohl sie die Verwaltung von Zugriffsrechten vereinfachen, basiert die tatsächliche Zugriffskontrolle weiterhin auf der Auswertung der zugrunde liegenden binären Berechtigungen. [12]

3.2.2.3. Scope von Berechtigungen

Ein zentrales Kriterium bei der Konzeption und Platzierung von Berechtigungen ist die Festlegung, welche Zugriffsrechte durch eine Berechtigung gewährt werden sollen. In diesem Zusammenhang kommt dem Grad der Differenzierung von Berechtigungen, der als Granularität bezeichnet wird, eine besondere Bedeutung zu. [12]

Die Grundlage für die Ausgestaltung von Berechtigungen bildet der Schutzbedarf der jeweiligen Ressource. Dieser wird üblicherweise im Rahmen einer Schutzbedarfsanalyse ermittelt, bei der unterschiedliche Einflussfaktoren berücksichtigt werden. Dazu zählen unter anderem die Relevanz der Ressource für Geschäftsprozesse sowie die Kritikalität ihrer Funktionen und Inhalte. [12]

Die ermittelten Schutzanforderungen fließen anschließend in die Berechtigungsplanung ein. Dabei wird festgelegt, auf welcher Detailebene der Zugriffsschutz umgesetzt werden muss und welche Funktionen durch entsprechende Berechtigungen abzusichern sind. Darüber hinaus wird bestimmt, wie stark der Zugriffsschutz ausgeprägt sein soll, beispielsweise ob der Einsatz erweiterter Authentifizierungsmechanismen erforderlich ist. [12]

3.2.2.4. Berechtigungsstufen

Anstatt jedem Nutzer komplizierte Einzelrechte zu geben, werden Berechtigungen in feste Stufen eingeteilt. Das macht die Verwaltung einfach: Man muss einem Nutzer nur noch eine **Ziffer** (die Stufe) zuweisen. [12] Die Skala reicht von *keine Berechtigung* bis *alle Berechtigungen*.

- **Stufe 0: Kein Zugriff**
  (Gar keine Rechte)
- **Stufe 1: Nur lesen**
  (Informationen ansehen)
- **Stufe 2: Lesen und Hinzufügen**
  (Daten ansehen und neue erstellen)
- **Stufe 3: Ändern und Löschen**
  (Bestehende Daten bearbeiten)
- **Stufe 4: Vollzugriff**
  (Alle Rechte)

3.2.3. Skalierbare plattformunabhängige Systemarchitekturen

3.2.3.1. Skalierbarkeit

Unter einer skalierbaren Systemarchitektur versteht man ein System, das sich problemlos an steigende Anforderungen anpassen lässt. Fehlende Skalierbarkeit führt dazu, dass Systeme bei wachsender Nutzerzahl oder zunehmendem Datenvolumen spürbare Performanceprobleme entwickeln. Während des Entwicklungsprozess sind unterschiedliche Prinzipien zu beachten:

Merkmal	**Beschreibung**
Modularität	Einzelne Komponenten der Software sollen unabhängig sein und sich flexibel aktualisieren oder erweitern lassen.
Flexibilität	Systeme müssen sich dynamisch an verändernde Anforderungen anpassen können.
Fehlertoleranz	Das System soll auf Fehler reagieren können und automatisch geeignete Maßnahmen ergreifen.

Tabelle 3.7.: Merkmale von Softwaresystemen

[13]

Um eine skalierbare Architektur zu gewährleisten, sind Optimierungsfunktionen von Bedeutung, um die Effizienz und Leistung zu optimieren. Wichtige Optimierungsstrategien sind unter anderem:

Technik	**Beschreibung**
Load Balancing	Verteilung von Anfragen auf mehrere Server, um Überlastung zu vermeiden.
Caching	Speicherung häufig abgerufener Daten, um Zugriffszeiten zu verkürzen.
Partitioning	Aufteilung von Daten in kleinere Einheiten, die parallel verarbeitet werden können.
Asynchrone Verarbeitung	Ermöglicht es Systemen, Aufgaben im Hintergrund auszuführen, was die Reaktionszeit verbessert.

Tabelle 3.8.: Techniken zur Leistungsoptimierung

[13]

Das Hauptziel einer skalierbaren Architektur besteht darin, auch bei steigender Nutzerzahl und wachsendem Datenvolumen eine hohe Verfügbarkeit sowie eine konsistent hohe Leistung sicherzustellen. Gleichzeitig muss das System so gestaltet sein, dass es leicht wartbar und problemlos erweiterbar bleibt, um zukünftige Anforderungen effizient integrieren zu können. Insgesamt ist eine durchdacht geplante skalierbare Architektur ein entscheidender Faktor für die langfristige Stabilität, Flexibilität und den Erfolg moderner Softwaresysteme. [13]

3.2.3.2. Plattformunabhängigkeit

Ein Computerprogramm benötigt eine Umgebung, in der es gestartet werden kann und während der gesamten Laufzeit stabil funktioniert. Ein Programm gilt als plattformunabhängig oder plattformübergreifend, wenn es auf verschiedenen Computersystemen ausgeführt werden kann, also auf Geräten mit unterschiedlicher Hardware, verschiedenen Prozessoren oder unterschiedlichen Betriebssystemen. Der Grad dieser Unabhängigkeit wird als Portierbarkeit (oder Portabilität) bezeichnet.

Die Portabilität kann z. B. geschätzt werden über

$$P = 1 - \frac{U + A}{E}$$

mit

- Übertragungsaufwand U (insbesondere Neukompilierung),
- Anpassungsaufwand A (Änderung des Quellcodes, z. B. bei Austausch von Betriebsschnittstellen),
- Entwicklungsaufwand E für Neuentwicklung.

Eine Portabilität von $P = 1$ bedeutet vollständige Kompatibilität; das Programm ist also ohne Änderungen auf dem Zielsystem lauffähig, was genau dann gilt, wenn $U = A = 0$.

Eine Quellcode-Portabilität liegt im Regelfall vor, wenn die Gesamt-Portabilität über 90% liegt. Dies entspricht einem Anpassungsaufwand von $A = 0$ und einem Übertragungsaufwand von $U < 0{,}1E$, da bei

$$P = 1 - \frac{U}{E} > 0{,}9$$

der Wert für U kleiner als ein Zehntel von E sein muss. Eine Portabilität nahe 0 entspricht hingegen einer nahezu vollständigen Neuentwicklung des Programms, wobei in diesem Fall $P \approx 0$ und somit $U + A \approx E$ gilt. [14]

Portabilität ist kein Maß für die Lauffähigkeit eines Programms auf der Zielplattform, d. h. selbst eine Portabilität von 99 % bedeutet nicht unbedingt, dass das Programm nutzbar ist, sondern lediglich, dass eine Portierung im Vergleich zu einer Neuentwicklung deutlich weniger Aufwand erfordert. Damit ist nicht nur gemeint, dass ein Programm auf mehreren Plattformen laufen kann, sondern auch, wie viel Aufwand nötig ist, um es dafür anzupassen. Dieser Vorgang wird Portierung oder Migration genannt. [14]

3.2.4. Sharepoint

3.2.4.1. Einordnung und Produkte

Microsoft SharePoint fungiert als webbasierte Kollaborationsplattform, die primär auf die Optimierung der teaminternen Zusammenarbeit sowie ein effizientes Informationsmanagement

ausgelegt ist. Durch die Bereitstellung zentraler Instanzen zur Inhaltsverwaltung ermöglicht die Plattform eine strukturierte Ablage und Verteilung von Daten. Ziel ist es, durch zusammenwirkende Kommunikation und medienbruchfreie Kooperation die organisatorische Produktivität zu steigern. [15]

Produkt	**Kernmerkmale und Fokus**
SharePoint in Microsoft 365	Cloudbasierte Plattform (SaaS) zur zentralen Inhaltsverwaltung und teamübergreifenden Freigabe von Dokumenten. Nutzung erfolgt im Rahmen von Microsoft-365-Abonnements.
SharePoint Server	On-Premise-Lösung für den Betrieb auf eigener IT-Infrastruktur. Ermöglicht volle Kontrolle über Datenhoheit und unterstützt hybride Szenarien mit der Cloud.
SharePoint Designer 2013	Spezialisiertes Werkzeug zur Erstellung deklarativer Workflows und zur Modellierung komplexer Geschäftsprozesse innerhalb der SharePoint-Umgebung.
OneDrive	Schnittstelle zur Synchronisation von SharePoint-Bibliotheken mit lokalen Endgeräten. Ermöglicht die Offline-Bearbeitung und Dateiverwaltung im lokalen Dateisystem.

Tabelle 3.9.: Überblick über SharePoint-Produkte

[15]

3.2.4.2. Architektur und Hierarchie

Die Struktur von SharePoint ist hierarchisch aufgebaut, um Inhalte logisch zu trennen und den Zugriff gezielt zu steuern. Den Kern dieser Architektur bildet die sogenannte Websitesammlung (Site Collection). [16]

Eine Websitesammlung besteht immer aus einer Website auf der obersten Ebene und allen darunterliegenden Websites. Sie dient als übergeordneter Container, in dem zentrale Einstellungen für die Sicherheit, das Design und verschiedene Funktionen festgelegt werden. Technisch gesehen landen alle Daten einer solchen Sammlung in einer einzigen Inhaltsdatenbank. Zwar kann eine Datenbank mehrere Websitesammlungen speichern, aber eine Sammlung kann nicht über mehrere Datenbanken verteilt werden. [16]

Bei der Planung der Struktur gibt Microsoft die Empfehlung ab, für jede Arbeitseinheit eine eigene Websitesammlung anzulegen, anstatt eine komplexe Verschachtelung mit vielen Unterwebsites zu bauen. Das hat den Vorteil, dass die Umgebung übersichtlich bleibt und später einfacher migriert werden kann. [16]

Innerhalb einer solchen Sammlung können Ressourcen wie Bilder, Vorlagen oder bestimmte Spaltentypen gemeinsam genutzt werden. Das sorgt dafür, dass die Navigation und das Erscheinungsbild für die Nutzer einheitlich bleiben. Bei der Benennung der URLs wird meistens auf pfadbasierte Adressen (wie z. B. „/sites/Projektname“) gesetzt, da diese mit Tools wie der PowerShell am einfachsten zu verwalten sind. [16]

3.2.4.3. Funktionale Elemente einer SharePoint-Website

Innerhalb einer Websitesammlung bestehen die einzelnen Websites aus verschiedenen funktionalen Elementen, die für die Organisation und Verwaltung der Daten zuständig sind:

- **Dokumentbibliotheken (Document Libraries):** Diese stellen den primären Speicherort für Dateien dar. Im Gegensatz zu herkömmlichen Dateiordnern ermöglichen sie die Nutzung von erweiterten Funktionen wie Versionierung, Check-Out-Mechanismen und die Verwendung von Metadaten. [17]
- **Listen (Lists):** Listen dienen der Erfassung strukturierter Daten, ähnlich einer Tabelle. Sie werden beispielsweise für Aufgabenlisten, Kontaktverzeichnisse oder Inventarlisten verwendet. [17]
- **Seiten (Pages):** Diese bilden das visuelle Interface der Website. Informationen werden hier mithilfe von sogenannten Webparts (funktionalen Bausteinen) für den Endanwender aufbereitet. [17]
- **Metadaten (Spalten):** Anstatt Dateien ausschließlich in starren Ordnerstrukturen zu sortieren, erlauben Spalten das Hinzufügen von Attributen wie „Dokumententyp“ oder „Status“. Dies verbessert die Filterbarkeit und Auffindbarkeit der Dokumente innerhalb der gemeinsamen Ablage erheblich. [17]

3.2.4.4. Kernfunktionen der Dokumentenverwaltung

SharePoint bietet eine Vielzahl von Funktionen, die speziell auf die Anforderungen der Dokumentenverwaltung zugeschnitten sind:

- **Versionierung:** Jede Änderung an einem Dokument wird als separate Version gespeichert. Dadurch können Nutzer frühere Versionen wiederherstellen oder Änderungen nachverfolgen. [17]
- **Auschecken und Einchecken (Check-Out/In):** Um Bearbeitungskonflikte zu vermeiden, können Dokumente exklusiv für einen Nutzer gesperrt werden. Während eine Datei ausgecheckt ist, können andere Nutzer diese zwar lesen, aber keine Änderungen vornehmen, bis sie wieder eingecheckt wird. [17]
- **Gemeinsame Dokumenterstellung (Co-Authoring):** Diese Funktion erlaubt es mehreren Anwendern, zeitgleich an demselben Dokument (z. B. Word oder Excel) zu arbeiten. Änderungen werden in Echtzeit synchronisiert, was die kollaborative Zusammenarbeit beschleunigt. [17]

3.2.5. Microsoft Entra ID

3.2.5.1. Begriff und Einordnung

Microsoft Entra beschreibt eine umfassende Produktfamilie für Identitätsmanagement und Netzwerkzugriff, die darauf ausgelegt ist, eine moderne Zero Trust-Sicherheitsstrategie in

Organisationen zu etablieren. Das Ziel dieser Architektur ist die Schaffung einer Vertrauensstruktur, die Identitäten sowie Zugriffsbedingungen und Berechtigungen konsequent überprüft, Verbindungskanäle verschlüsselt und eine kontinuierliche Überwachung auf Kompromittierungen ermöglicht. Die hierarchische Struktur und die verschiedenen Säulen dieser Architektur sind in Abbildung 3.2 dargestellt. [18]

Das zentrale Kernprodukt dieser Familie ist Microsoft Entra ID. Dabei handelt es sich um einen cloudbasierten Dienst zur Identitäts- und Zugriffsverwaltung, der die grundlegenden Funktionen für Authentifizierung und den Schutz von Benutzern, Geräten sowie Anwendungen bereitstellt. Eine wesentliche Besonderheit für Unternehmen im Microsoft-Umfeld besteht darin, dass Abonnenten von Diensten wie Microsoft 365 oder Azure automatisch einen Microsoft Entra-Mandanten nutzen und somit direkt mit der Verwaltung ihrer Cloud-Anwendungen beginnen können. Über das webbasierte Microsoft Entra Admin Center lassen sich diese Produkte zentral über eine einzige Benutzeroberfläche konfigurieren und verwalten. [18]

Abbildung 3.2.: Übersicht der Microsoft Entra Produktfamilie [18]

3.2.5.2. Zentrale Identitätsverwaltung

Die zentrale Verwaltung innerhalb von Microsoft Entra sorgt dafür, dass Identitäten für unterschiedliche Nutzergruppen und Ressourcen an einem einzigen Ort kontrolliert werden. Dies umfasst nicht nur menschliche Identitäten wie Mitarbeitende, Partner oder Kundschaft, sondern schließt auch Geräte sowie sogenannte Workload-Identitäten (z. B. Anwendungen und Dienste) mit ein. [18]

Ein wesentlicher Bestandteil dieser Verwaltung ist die Microsoft Entra ID Governance. Diese ermöglicht es, den gesamten Lebenszyklus einer Identität zu automatisieren. So können beispielsweise Zugriffsanfragen, Zuweisungen von Lizenzen und regelmäßige Überprüfungen von Berechtigungen systemgesteuert erfolgen. Dies stellt sicher, dass neuen Mitarbeitern automatisch die benötigten Ressourcen zugewiesen werden und diese Zugänge beim Verlassen des Unternehmens ebenso zuverlässig wieder entfernt werden. [18]

Für die Verwaltung externer Nutzer bietet Microsoft Entra External ID eine sichere Methode zur Zusammenarbeit. Hiermit können Geschäftspartner oder Gäste gezielt Zugriff auf interne Ressourcen erhalten, während Kunden über Self-Service-Registrierungen (z. B. via Google- oder Facebook-Konten) angebunden werden können. [18]

3.2.5.3. Authentifizierung und Autorisierung

Innerhalb von Microsoft Entra ID bilden Authentifizierung und Autorisierung die Grundlage für den sicheren Zugriff auf Ressourcen, folgen jedoch unterschiedlichen Prinzipien:

- **Authentifizierung (Identity):** Dies ist der Prozess der Identitätsprüfung. Hierbei stellt das System sicher, dass ein Benutzer tatsächlich die Person ist, für die er sich ausgibt. Dies geschieht in der Regel durch die Abfrage von Anmeldedaten oder sicheren Methoden wie der Multi-Faktor-Authentifizierung (MFA). Microsoft Entra ID Protection unterstützt diesen Vorgang, indem es risikobasierte Richtlinien nutzt, um verdächtige Anmeldeversuche automatisch zu erkennen und zusätzliche Bestätigungen einzufordern. [18]
- **Autorisierung (Permissions):** Nach einer erfolgreichen Anmeldung bestimmt die Autorisierung, welche spezifischen Zugriffsrechte der Benutzer besitzt. Es wird festgelegt, auf welche Anwendungen, Dateien oder Funktionen (z. B. innerhalb einer SharePoint-Website) zugegriffen werden darf. [18]

3.2.5.4. Single Sign-On (SSO)

Ein zentrales Leistungsmerkmal von Microsoft Entra ID ist das sogenannte Single Sign-On. Diese Funktion ermöglicht es Benutzern, sich mit nur einer einzigen Identität – bestehend aus einem Benutzernamen und einem Kennwort – einmalig anzumelden, um danach automatisch Zugriff auf alle für sie freigegebenen Cloud-Anwendungen sowie lokalen Ressourcen zu erhalten. [18]

Aus technischer Sicht bietet SSO zwei wesentliche Vorteile für das Unternehmen:

- **Benutzerfreundlichkeit:** Da sich die Anwender nicht mehr für jede einzelne Anwendung (wie SharePoint, Teams oder externe Drittanbieter-Tools) unterschiedliche Zugangsdaten merken müssen, wird die tägliche Arbeit effizienter. Zudem sinkt die Anzahl an Support-Anfragen aufgrund vergessener Passwörter erheblich. [18]
- **Sicherheit:** Da die Identität zentral an einer Stelle geprüft wird, können Sicherheitsrichtlinien, wie beispielsweise die Mehrfaktor-Authentifizierung (MFA), global für alle angebundenen Programme erzwungen werden. Zudem wird das Risiko minimiert, dass Nutzer unsichere Passwörter für mehrere Dienste gleichzeitig verwenden oder diese ungesichert notieren. [18]

3.2.5.5. Integration externer Anwendungen

Ein wesentlicher Vorteil von Microsoft Entra ID ist die Fähigkeit, über die Grenzen der Microsoft-Welt hinaus als zentrale Identitätsinstanz zu fungieren. Dies wird durch die Integration externer Anwendungen (Third-Party-Apps) realisiert:

- **SaaS-Anwendungen:** Über vorkonfigurierte Schnittstellen lassen sich gängige Cloud-Dienste (wie Salesforce, Slack oder Dropbox) nahtlos anbinden. Dadurch profitieren auch diese externen Dienste von Sicherheitsfeatures wie Single Sign-On (SSO) und der Mehrfaktor-Authentifizierung. [18]
- **Zentrale App-Verwaltung:** Administratoren können im Entra Admin Center exakt steuern, welche Benutzer oder Gruppen Zugriff auf bestimmte externe Software erhalten. Dies vereinfacht das Onboarding neuer Mitarbeiter, da Zugänge für verschiedene Plattformen an einer zentralen Stelle zugewiesen werden können. [18]
- **Sicherer Internet- und Privatzugriff:** Moderne Komponenten wie der *Microsoft Entra Internet Access* und *Private Access* erweitern diesen Schutz. Sie ermöglichen einen sicheren Zugriff auf Internet-Ressourcen sowie interne Unternehmensnetzwerke, ohne dass klassische, oft unsicherere VPN-Verbindungen notwendig sind. [18]

Die zentrale Rolle von Microsoft Entra ID als Vermittler zwischen lokalen Infrastrukturen, Cloud-Applikationen und externen Identitäten wird in Abbildung 3.3 verdeutlicht. Durch diese weitreichende Integrationsfähigkeit wird Microsoft Entra ID zum zentralen Kontrollpunkt für die gesamte IT-Infrastruktur einer Organisation, was die Sicherheit erhöht und die Komplexität für die Endanwender reduziert.

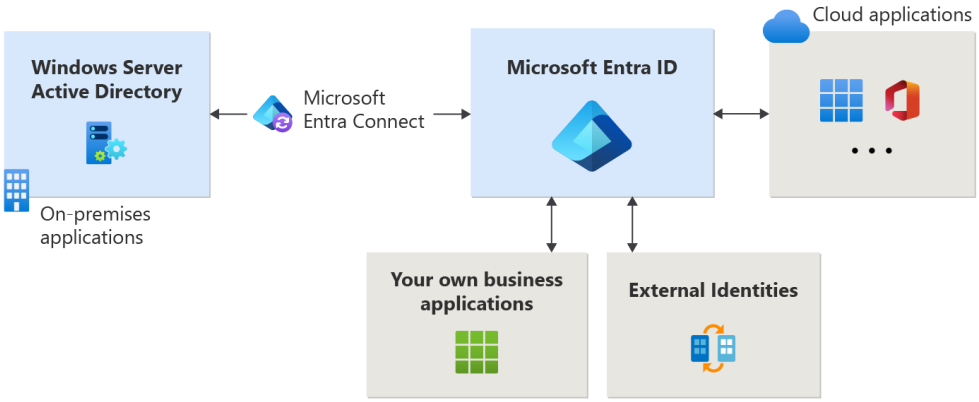

Abbildung 3.3.: Integration von On-Premises- und Cloud-Anwendungen über Microsoft Entra ID [18]

4. Dokumentation der Implementierung

4.1. Dokumentation - Grundlegend

4.1.1. Test Umgebung

4.1.2. Technologien

4.2. Dokumentation - Funktionen

4.2.1. Dokumenten-Upload

4.2.2. Dokumenten-Klassifikation

4.2.3. Dokumenten-Suche

4.2.4. Benutzer- und Rollenverwaltung

4.2.5. System-Logging und Monitoring

4.2.6. API-Endpunkte

4.2.7. Fehlerbehandlung und Ausnahmen

4.2.8. Sicherheitsfunktionen

5. Beurteilung

5.1. Bewertung der Implementierung

5.2. Erfüllung der Forschungsfragen

5.3. Kritische Reflexion und Grenzen

6. Zusammenfassung und Ausblick

6.1. Zusammenfassung

Zusammenfassend war diese Diplomarbeit ein sehr lehrreiches Projekt, bei dem wir viele neue Erfahrungen gemacht haben. ...

6.2. Ausblick

I. Literaturverzeichnis

[1] SER Group: *Document Classification – mit KI zum optimalen Input Management*, Januar 2024. Online in Internet: URL: `https://www.sergroup.com/de/knowledge-center/blog/document-classification-mit-ki.html`.

[2] Shaip: *KI-basierte Dokumentenklassifizierung – Vorteile, Prozesse und Anwendungsfälle*, Juli 2025. Online in Internet: URL: `https://de.shaip.com/blog/ai-based-document-classification/`.

[3] Evoluce GmbH: *Merkmalsextraktion mit KI: Relevantes erkennen, Unwichtiges ausblenden.* Online in Internet: URL: `https://evoluce.de/merkmalsextraktion/` (Zugriff: 03.02.2026). Veröffentlichungsdatum auf der Seite nicht angegeben.

[4] Barnard, Joel: *Was ist Embedding?* Online in Internet: URL: `https://www.ibm.com/de-de/think/topics/embedding` (Zugriff: 03.02.2026).

[5] Dascalescu, Dan and Zain Hasan: *Vector embeddings explained*, January 2023. Online in Internet: URL: `https://weaviate.io/blog/vector-embeddings-explained` (Zugriff: 03.02.2026).

[6] Elastic: *Text classification.* Online in Internet: URL: `https://www.elastic.co/de/what-is/text-classification` (Zugriff: 03.02.2026).

[7] Google Cloud: *What are convolutional neural networks (cnns)?* Online in Internet: URL: `https://cloud.google.com/discover/what-are-convolutional-neural-networks` (Zugriff: 03.02.2026).

[8] Harley, Adam: *Rvl-cdip: A large-scale dataset for document classification.* Online in Internet: URL: `https://adamharley.com/rvl-cdip/` (Zugriff: 03.02.2026).

[9] V7 Labs: *Vision transformers (vit): A practical guide.* Online in Internet: URL: `https://www.v7labs.com/blog/vision-transformer-guide` (Zugriff: 03.02.2026).

[10] Hugging Face: *Layoutlm.* Online in Internet: URL: `https://huggingface.co/docs/transformers/en/model_doc/layoutlm` (Zugriff: 03.02.2026).

[11] Azure Samples: *Document classification with gpt vision (notebook).* Online in Internet: URL: `https://github.com/azure-samples/azure-ai-document-processing-samples/blob/main/samples/python/classification/document-classification-gpt-vision.ipynb` (Zugriff: 03.02.2026).

[12] Tsolkas, Alexander und Klaus Schmidt: *Rollen und Berechtigungskonzepte: Identity-und Access-Management im Unternehmen.* Springer-Verlag, 2017.

[13] StudySmarter: *Skalierbare Architekturen: Definition, Arten & Beispiele*, 2024. Online im Internet: URL: `https://www.studysmarter.de/schule/informatik/technische-informatik/skalierbare-architekturen/`, abgerufen am 27. Januar 2024.

[14] Wikipedia-Autoren: *Plattformunabhängigkeit — Wikipedia, Die freie Enzyklopädie*, Dezember 2024. Online im Internet: URL: `https://de.wikipedia.org/wiki/Plattformunabh%C3%A4ngigkeit`, abgerufen am 27. Januar 2026.

[15] Microsoft Support: *SharePoint – Hilfe und Lernen*, 2024. `https://support.microsoft.com/de-de/sharepoint`, Zugegriffen am: 02.02.2026.

[16] Microsoft Learn: *Übersicht über Websites und Websitesammlungen in SharePoint Server*, 2023. `https://learn.microsoft.com/de-de/sharepoint/sites/sites-and-site-collections-overview`, Zugegriffen am: 02.02.2026.

[17] Microsoft Corporation: *SharePoint – Hilfe und Lernen*, 2026. `https://support.microsoft.com/de-de/sharepoint`, Zugriff am 25. Februar 2026.

[18] Microsoft: *Was ist Microsoft Entra?*, 2025. `https://learn.microsoft.com/de-de/entra/fundamentals/what-is-entra`, Abgerufen am 2. Februar 2026.

[19] abc: *DB-Engine Ranking*, März 2016. Online in Internet: URL: `http://db-engines.com/de/ranking`.

[20] Griesch, Leon, Leon Görgen und Kurt Sandkuhl: *Ki-als-service: Vergleich von plattformen zur dokumentenklassifikation.* In: *INFORMATIK 2024*, Seiten 1505–1518. Gesellschaft für Informatik eV, 2024.

II. Abbildungsverzeichnis

3.1. Übersicht der Vektordarstellung von Embeddings [5] 15
3.2. Übersicht der Microsoft Entra Produktfamilie [18] 28
3.3. Integration von On-Premises- und Cloud-Anwendungen über Microsoft Entra ID [18] 30

III. Tabellenverzeichnis

3.1. Textklassifizierung vs. Dokumentenklassifizierung [2] 11
3.2. Wichtige Begriffe der Merkmalsextraktion [3] 13
3.3. Relevante Algorithmen zur Merkmalsextraktion [3] 14
3.4. Relevante Objekte für Embeddings [4] 17
3.5. Vergleich zwischen lexikalischer und semantischer Suche 21
3.6. Komponenten einer Berechtigung 22
3.7. Merkmale von Softwaresystemen 24
3.8. Techniken zur Leistungsoptimierung 24
3.9. Überblick über SharePoint-Produkte 26

A.1. Kapitelverzeichnis 39
A.2. Arbeitstagebuch Toifl 39
A.3. Arbeitstagebuch Schaider 39

IV. Quellcodeverzeichnis

A. Anhang

A.1. Arbeitsteilung

Kurze Beschreibung, wer was gemacht hat (Überblick).

A.2. Kapitelverzeichnis

Kapitel	Editor
?? ??	Max Mustermann
?? ??	Mex Musterjuan

Tabelle A.1.: Kapitelverzeichnis

A.3. Projekttagebücher

A.3.1. Projekttagebuch Max Mustermann

Tag	Zeit	kumulativ	Fortschritt
Mo 28.11.16	2h	2h	Besprechnung der Programmanforderungen
Di 29.11.16	3h	5h	Datenbankmodell erstellt
Mi 30.11.16	1h	6h	Datenbankmodellüberarbeitet
Do 01.12.16	3h	9h	Pflichtenheft erstellt

Tabelle A.2.: Arbeitstagebuch Toifl

A.3.2. Projekttagebuch Mex Musterjuan

Tag	Zeit	kumulativ	Fortschritt
Mo 28.11.16	2h	2h	Besprechnung der Programmanforderungen

Tabelle A.3.: Arbeitstagebuch Schaider

A.4. Besprechungsprotokolle

. . . Hier können auch pdf Dateien eingebunden werden!

htl bildung mit zukunft	**HTL Krems** **Höhere Lehranstalt für Informationstechnologie** **Ausbildungsschwerpunkt**	**Reife- und Diplomprüfung**

Betreuungsprotokoll zur Diplomarbeit lfd. Nr.:

Themenstellung:
Kandidaten/Kandidatinnen:

Jahrgang:
Betreuer/in:
Ort:
Datum:
Zeit:

Besprechungsinhalt:

Name	Notiz

Aufgaben:

Name	Notiz	zu erledigen bis

A.5. Datenträgerbeschreibung