Hochschule **RheinMain**
Fachbereich Design Informatik Medien
Studiengang Medieninformatik

Abschlussarbeit

zur Erlangung des akademischen Grades

Master of Science

Data Mining komplexer Datenstrukturen aus PDF-Dokumenten

Vorgelegt von	Deniz Aydar
am	25. Juli 2022
Referent	Prof. Dr. Dirk Krechel
Korreferent	Prof. Dr. Philipp Schaible

Erklärung gemäß ABPO

Ich erkläre hiermit, dass ich

- die vorliegende Abschlussarbeit selbstständig angefertigt,
- keine anderen als die angegebenen Quellen benutzt,
- die wörtlich oder dem Inhalt nach aus fremden Arbeiten entnommenen Stellen, bildlichen Darstellungen und dergleichen als solche genau kenntlich gemacht und
- keine unerlaubte fremde Hilfe in Anspruch genommen habe.

Wiesbaden, 25. Juli 2022 _____

Deniz Aydar

Erklärung zur Verwendung der Masterthesis

Hiermit erkläre ich mein Einverständnis mit den im folgenden aufgeführten Verbreitungsformen dieser Abschlussarbeit:

Verbreitungsform	**Ja**	**Nein**
Einstellung der Arbeit in die Hochschulbibliothek mit Datenträger		×
Einstellung der Arbeit in die Hochschulbibliothek ohne Datenträger		×
Veröffentlichung des Titels der Arbeit im Internet	×	
Veröffentlichung der Arbeit im Internet		×

Wiesbaden, 25. Juli 2022 _____

Deniz Aydar

Zusammenfassung

PDF-Dokumente besitzen viele Informationen aus denen sich neue Daten generieren lassen können. Doch das Extrahieren von solchen Daten ist heutzutage immer noch mit Hürden verbunden. Dies gilt auch für die Dokumente die in den Prozessen von Autohäusern und Kfz-Werkstätten verwendet und anschließend gelagert werden. Um die Frage zu beantworten inwiefern neue Daten aus diese Art von Dokumenten verarbeiten lassen, wird im Rahmen dieser Arbeit ein System konzipiert und entwickelt welches es ermöglichen soll weiteres Wissen zu gestalten.

Inhaltsverzeichnis

Kapitel 1

Einleitung

„Digitalisierung im Alltag voranbringen“ – Das war einer der Wahlslogans während der Bundestagswahl 2021. Gemeint war damit eine zunehmende Digitalisierung im privaten Alltag vieler Bürger:innen, aber auch in der Wirtschaft machte sich wachsend der Wunsch nach mehr digitalen Alternativen breit (vgl. Bundesregierung 2021). Dieser Wunsch beinhaltete vor allem einen Wechsel von gängigen Papierformen verschiedenster Dokumente zu denselben in digitaler Ausprägung.

Genau jenes Bedürfnis nach Digitalisierung betrifft auch Arbeitnehmer:innen / Händler:innen in Autohäusern und Kfz-Werkstätten, die durch ihre berufliche Tätigkeiten mit einer Vielzahl an Unterlagen, Dokumenten oder Belegen arbeiten müssen. Unter dieser Vielzahl fallen Dokumente wie Werkstatt-, Kauf- und Mietverträge sowie Rechnungen oder Diagnoseberichte.

Durch eine Digitalisierung jener Dokumente eröffnen sich Vorteile wie bessere Zugänglichkeit, größere Langlebigkeit und vor allem eine angepasste und leichtere Nutzung, die zu einer höheren Effektivität innerhalb täglicher Arbeitsschritte führt. Durch bisherige erste Schritte der Digitalisierung sind diese benötigten Dokumente bereits elektronisch aufgearbeitet und den Arbeitnehmenden in Autohäusern und Kfz-Werkstätten zur Verfügung gestellt worden.

Die Folge dessen ist, dass alle Dokumente standardisiert sind und dadurch die in den Dokumenten beinhalteten Informationen neu verarbeitet werden können. Eine Extraktion der Daten der einzelnen Dokumente ist jedoch nur eingeschränkt möglich, da die Struktur im gängigen Gebrauch im Dateiformat PDF (Portable Document Format) festgelegt ist.

Jene Limitierung der Datenextraktion wirkt sich gleichermaßen auf die Dealer-Management-Systeme (DMS), mit solchen die Arbeitnehmenden ihr Prozesse

abwickeln, aus. Der daraus resultierende Arbeitsaufwand, welcher dabei entsteht, um die Daten wiederverwendbar zu konstruieren, ist derzeit enorm.

1.1 Zielsetzung

Aus diesem Grund möchte ich innerhalb dieser Master-Thesis ein System entwickeln, das genau jene Problematik erleichtert und löst. Innerhalb üblicher Methoden werden heutzutage die Daten zunächst über eine grafische Anzeige mithilfe einer bereits existierenden Benutzeranwendung via gängiges Kopieren und Einfügen entnommen. Dieser Schritt funktioniert zwar im Regelfall, ist jedoch oftmals stark fehleranfällig und kann lückenhaft sein.

In so einem Fall muss das Einfügen und Kopieren manuell durchgeführt werden, was bei einer riesigen Menge an Dokumente zu einer monotonen Arbeit führt. Andernfalls können die Daten eigenhändig abgeschrieben werden – jener Vorgang benötigt jedoch viele Ressourcen. Eher geeignet ist es, einen Datenkonverter für die PDF Dateien zu nutzen, um so die Dokumente in Bilder beispielsweise umzuwandeln, wodurch die Daten zugänglicher sind, aber immer noch verarbeitet werden müssen, was den gesamten Vorgang als eine Notlösung wirken lässt.

Aufgrund dieser bisherigen teils aufwendigen und mit Fehlern verbundenen Möglichkeiten möchte ich mich in meiner Thesis von diesen Optionen abwenden und das Data-Mining nutzen. Das Data-Mining soll eine Automatisierung unterstützen, mit der Daten aus dem PDF-Dokument extrahiert werden. Allein über das Data-Mining ist es möglich, Inhalte zu extrahieren und für andere Systeme bereit zu stellen.

Der Unterschied zum gängigen Data-Mining liegt dabei in den Einschränkungen des Dateiformats: der Quelltext einer solchen Datei stellt erstens keine klare Hierarchie der Daten dar und zweitens besitzt es durch das Fehlen von Markierungen keine Informationen darüber, was die Daten an sich darstellen sollen [8].

Mit Hilfe dieser Technik des Data-Minings soll es ermöglicht werden, aktuelle Prozesse detaillierter zu steuern und zu überwachen. Des Weiteren können unter anderem Abgleiche von Rechnungen mit dem System durchgeführt werden und weitere Datenanalyse und Reporting kreiert werden. Mit dem derzeitigen Stand der Datenextraktion können jedoch lediglich Agierende aus dem technischen Bereich arbeiten. Die Technologien hierfür benutzen unterschiedliche Ansätze und müssen daher zunächst für diesen Fall ausgewählt werden.

Damit jedoch auch für Agierende innerhalb des technischen Bereichs die Nutzung nicht zu abstrakt bleibt, soll durch die Entwicklung eines Systems der Zugriff greifbarer gestaltet werden, welches wiederum eine benutzerfreundliche Anwendung zur Verfügung stellen soll. Darüber hinaus muss eine Automatisierung vorhanden sein, um auch eine große Datenmenge verarbeiten zu können, und die Vorgänge sollen außerdem eine niedrigere Fehlerquote aufzeigen. Die Prozesse müssten außerdem das ganze über eine Steuerung und Regelung kontrollieren. Solche Prozesse können zum Beispiel über Regeln festgelegt werden, die bei Erfüllung weitere Aktionen auslösen können.

Eine solche Eigenschaft kann durch eine symbolische künstlichen Intelligenz (KI) abgedeckt werden, welche eine vorgegebene Verarbeitung möglich macht. Hiermit bezeichnet man einen altmodischen Ansatz der KI, bei der über das Festlegen von Symbolen menschliches Wissen in einer Logik gehalten wird und diese benutzt wird um weiteres Wissen zu generieren [7]. Durch eine Parametrisierung einer solchen KI kann dann die Unschärfe der ausgewählten Daten festgelegt werden, damit eine Feineinstellung stattfinden kann. Das heißt, der Benutzer eines solchen Systems soll sich beginnend von groben Definitionen für die Verarbeitung zu einer detaillierten und angepassteren Definition über Feedback des Systems hinarbeiten.

Mit Hilfe dieser Annäherung an Definitionen und Regeln kann so für eine bessere Erfolgsquote gesorgt werden. Zudem kann der Ansatz des Machine Learnings (ML), wobei erfolgreiche Verarbeitungen einem System antrainiert werden, weitere Dokumente aus Daten extrahieren. Hierbei ist noch offen, welcher Typ des Machine Learnings sich für diesen Fall eignet.

Beide Ansätze – der der symbolischen KI und der des Machine Learnings – bieten als Option die Entwicklung einer grafischen Entwicklungsumgebung (IDE) an. Jene soll das Festlegen von Definitionen und Regeln erlauben, mit welchen die Dokumente verarbeitet werden können. Außerdem soll diese Oberfläche verschiedene Funktionalitäten anbieten und auch Feedback sowohl bei einem problemlosen Lauf als auch bei einem fehlerhaften Lauf zurückgeben.

Des Weiteren soll das System es ermöglichen Änderungen der Definitionen anzumerken und Unterschiede zu erstellen, womit auch bisherige Ergebnisse angezeigt werden. Die Festlegung von Definitionen und Regeln soll außerdem durch eine Ansicht der Dokumente unterstützt werden. Insgesamt wird das System die zwei Ansätze als Subsysteme aufteilen um die Umsetzung zu modularisieren und einen Vergleich zu ermöglichen.

1.2 Ablauf

Um genau dieses optimale System für das beschriebene Szenario entwickeln zu können, werde ich in dieser Arbeit wie folgt vorgehen:

Für eine vollständige Auseinandersetzung soll eine weitere Analyse stattfinden, damit die vorhandenen Grundlagen für den Anwendungsfall evaluiert werden können. Das heißt, es werden mit Hilfe der Funktionalitäten der Bibliotheken Ergebnisse auf Basis der Datensätze erzeugt, um die Erfolgsquoten zu überprüfen. Diese werden dann verglichen und ausgewertet. Parallel dazu soll die Umgebung für die Entwicklung eingerichtet werden, mit der die Implementierung der Systeme stattfinden soll.

Danach widme ich mich der Konzeption und der Entwicklung des Systems. Dies beginnt mit der symbolischen KI als ein Subsystem des gesamtem Projekts, sodass ein direkterer Ansatz ermöglicht wird. Die Entwicklung hierbei wird im Wasserfall-Ansatz, einem Ansatz bei der phasenweise die Eigenschaften einer solchen KI umgesetzt werden, um eine Grundlage für die nächste Komponente zu bieten.

Bei dieser Komponente handelt es sich um die IDE, die den Zugang für den Benutzer zum System darstellt. Deswegen folgt im nächsten Schritt die Konzeptionierung und Entwicklung der IDE, welche sich während der Entwicklung der symbolischen KI parallelisieren lässt. Sobald das Gerüst der Benutzeroberfläche fertiggestellt ist, möchte ich dies mit der symbolischen KI verbinden.

Als Gegenstück wird sich dann mit dem Machine Learning Subsystem, bei der auch eine Konzeptionierung und Entwicklung, gewidmet. Hier wird das ganze in einem iterativen Ansatz umgesetzt um so den Hauptteil so früh wie möglich parat zu haben, damit das Training des ML-Systems durch die Dokumente auch zeitnah starten kann. Durch das Auswerten dieses Subsystems wird es daraufhin angepasst.

Danach wird das Testing für beide Subsysteme eingeführt um so das erfolgreiche Ausführen gewisser Eigenschaften abzudecken. Dies dient wiederum als Grundlage um die Continuous Integration (CI) für das System einzuführen, womit eine sichere und konsistente Entwicklung nach dem Prototyping angeboten wird.

Wenn dann aber die Systeme gegeben sind, kann das grafische Tool weiter angepasst werden welches sich dann mit den Systemen verbinden soll um die Ausführbarkeit der Funktionalitäten zu überprüfen. Hier werden dann die Komponenten zusammengeführt um ein gemeinsames System zu ermöglichen, welches dann gegebenenfalls weitere Testschritte bedarf. Sobald diese eine erfolgreiche Form annimmt kommt der Teil der Continuous Integration hinzu,

womit das System einen Zustand der Betrieblichkeit annehmen kann. Über Testing sollen so mögliche Fehlerquellen entdeckt werden bevor diese auftreten können.

Unter die Zielsetzung für die beiden Ansätze fallen die Kundendaten mit den Eigenschaften wie Name, Firma, Adresse und Kundennummer, die Fahrzeugdaten worunter Fahrgestellnummer, Modell, Farbe, Motor, Erstzulassung, Letzter/-nächster Service und TÜV fallen und die Abrechnungsdaten wie die Jobs mit Arbeitspositionen und Teilen, Preise etc. Dazu stehen als weitere Grundlage mehr als 10.000 annotierte unterschiedliche Dokumente mit genau den Daten, die extrahiert werden sollen als Datenbank zur Verfügung. Für das Mengengerüst sollen mehr als 100.000 Dokumente pro Jahr später bearbeitet werden. Hierbei ist es aber auch möglich aus den bereits bestehenden Daten einen Pool für den Lernprozess bzw. für die Verifikation zu bilden.

Zum Schluss werden sich die Ergebnisse des gesamten Systems betrachtet und ein Ausblick zu der Thematik dargestellt.

Kapitel 2

Hintergrund

Um die Problematik detaillierter darzustellen und um eine Übersicht zu ermöglichen, werden in diesem Kapitel der Rahmen der Problematik aus technischer Sicht vertieft dargestellt. Hierbei sollen die Technologien abgewogen und evaluiert werden.

2.1 Analyse

Das ERP System erzähl von der struktur und wie die Dokumente gehalten werden....

Das Extrahieren von Daten ist eine gängige Methode um Informationen aus verschiedenen Systemen wie Datenbanken oder Software as a Service (SaaS) Plattformen zu beschaffen [10]. Durch diese Beschaffung können die Daten weiter verarbeitet werden um neue Lösungsmöglichkeiten für das eigene System anzubieten. Die Arten des Extrahierens unterscheiden sich hierbei im Zeitverlauf und der Herangehensweise. So können Diese extrahiert werden sobald Änderungen der Daten beobachtet oder mitgeteilt werden, wodurch die Daten nach und nach beschafft werden.

Des Weiteren gibt es die Option eine komplette Extraktion durch zu führen. Dieser Ansatz kann sich in vielen Fällen als nachteilig erweisen, da die Daten jedes mal bei Anpassungen der ursprünglichen Information neu verarbeitet werden müssen. Da sich die Problematik im Rahmen der Thesis auf Dokumente, die bereits im ERP-System archiviert wurden, bezieht, trifft dieser Fall hier nicht zu.

Für das Data Mining von PDF Dateien existieren bereits Lösungen, die von großen Unternehmen wie Amazon Web Services (AWS) oder Adobe bereits

angeboten werden. So nutzen Produkte wie Textractor [2] von AWS oder Adobes Extractor API [6] Künstliche Intelligenz, Machine Learning und Optical Character Recognition (OCR) um die Extraktion der Daten zu ermöglichen. Unternehmen wie ABBYY [1] habe sich bereits auf diesem Gebiet spezialisiert und sind erfolgreich mit der Datenextraktion. Auch existieren bereits Open-Source Bibliotheken, die sich mit dem Entnehmen der Daten aus PDF Dokumenten beschäftigen.

Die Meisten davon lassen sich hinsichtlich der Aufteilung der Aufgaben gruppieren. So erlauben unter anderem PDFMiner [12] oder Apache Tika [3] Texte aus PDF Dokumenten zu gewinnen. Funktionalitäten für das Extrahieren von Daten aus Tabellen bieten Projekte wie Tabula [11] oder Camelot [5] an womit ein weiterer Aspekt durch die Nutzung abgedeckt wäre.

Bei Beiden zeigen sich niedrige Fehlerquoten auf [4] und die Nutzung von einer Fuzzy Logik. Camelot ermöglicht dazu noch ein Web Interface womit die simple Extraktion über einen Browser lokal stattfinden kann. Andere Bibliotheken wie Textricator [9] sind umfangreicher gestaltet und ergeben geeignete Schnittstellen für die Anfragen spezifischer Inhalte. Doch auch wenn dies mit textbasierten Dokumenten funktioniert und der Export flexibel ist, können weder nicht-textbasierte noch komplexere PDF Dokumente verarbeitet werden. Des Weiteren sind die Verarbeitungen nicht automatisierbar und die Nutzergruppe wird durch eine fehlende grafische Komponente eingeschränkt.

Kapitel 3

Konzeption und Umsetzung

Für den Sprung zur Umsetzung bedarf es als Nächstes die Konzeption des technischen Rahmens.

Kapitel 4

Ergebnisse

Kapitel 5

Ausblick und Fazit

Im Rahmen der Abschlussarbeit wurde technische Funktionalitäten eingeführt werden um dem Kapitän die Webanwendung als PWA anzubieten.

Insgesamt eignen sich Webanwendungen um Ressourcen übersichtlicher darzustellen und mobil zu beobachten. Auch können Offline-Zustände dem Benutzer trotzdem erlauben Aspekte der Anwendung weiter zu benutzen. Aus den Konzepten und der Umsetzung ergeben sich viele weitere Ideen und Ansätze, die realisierbar sind und implementiert werden können.

Literaturverzeichnis

Online-Quellen

[1] abbyy. PDF Software: Open, Read & Edit PDFs.

[2] amazon. Intelligente Extraktion von Text und Daten mit OCR – Amazon Textract – Amazon Web Services.

[3] apache. Apache Tika – Apache Tika.

[4] atlanhq. Comparison with other PDF Table Extraction libraries and tools · atlanhq/camelot Wiki.

[5] camelot. Camelot: PDF Table Extraction for Humans, July 2022. original-date: 2016-06-18T11:48:49Z.

[6] Adobe I/O-Adobe Developers. Extract Text from PDF | Extract Data from PDF | Visualizer - Adobe Developers.

[7] Ben Dickson. What is symbolic artificial intelligence?, November 2019.

[8] Docsumo. PDF Scraper - Scrape data from pdf | PDF data extraction, June 2022.

[9] measures. measuresforjustice/textricator: Textricator is a tool to extract text from documents and generate structured data.

[10] stichdata. What is Data Extraction? [Tools & Techniques].

[11] tabulapdf. tabulapdf/tabula: Tabula is a tool for liberating data tables trapped inside PDF files.

[12] unixuser. PDFMiner.