

TextStrip2.0

Projektauftrag, Lasten- und Pflichtenheft

Autoren: Michel Medved, Jan Neifeld, Tobias Kister, Daniel Schmitz

Mannheim, den 24.05.2023

Versionsübersicht:

Version	Datum	Status	Änderung
0.1	24.05.2023	In Prüfung	Initiale Fassungen des Projektauftrags wie auch des Lasten- und Pflichtenhefts

1 INHALTSVERZEICHNIS

1	Inhaltsverzeichnis	2
2	Projektauftrag.....	3
2.1	Projektstammdaten	3
2.2	Beschreibung der Ausgangssituation	3
2.3	Definition der Projektziele.....	3
2.4	Projektabgrenzung	5
2.4.1	In-Scope.....	5
2.4.2	Out-of-Scope	5
2.5	Risikoanalyse	6
2.5.1	Risiken während des Projektes	6
2.5.2	Risiken bei Nichtdurchführung des Projektes	6
2.6	Phasen und Projektabhängigkeiten	7
2.7	Projektorganisation	8
3	Lastenheft.....	9
3.1	Einleitung.....	9
3.1.1	Allgemeines	9
3.1.2	Meeting-Protokolle	9
3.2	Konzept und Rahmenbedingungen	10
3.2.1	Benutzer / Zielgruppe.....	10
3.2.2	Ziele und Nutzen für den Anwender	10
3.2.3	Systemvoraussetzungen und Ressourcen	10
3.3	Anforderungsbeschreibung	11
3.3.1	Funktionale Anforderungen	11
3.3.2	Nicht funktionale Anforderungen	16
3.3.3	Abnahmekriterien	17
4	Pflichtenheft.....	18
4.1	Einleitung.....	18
4.1.1	Allgemeines	18
4.1.2	Meeting-Protokolle	18
4.2	Ressourcen	18
4.3	Meilensteine.....	19
4.4	Umsetzung der vordefinierten Anforderungen	19
4.4.1	Funktionale Anforderungen	19
4.4.2	Nicht funktionale Anforderungen	26

2 PROJEKTAUFTRAG

2.1 Projektstammdaten

Projektname:	Textstrip2.0	Projektnummer:	0001
Projekttyp:	Studienprojekt zur Verringerung des Zeitaufwandes bei der Recherche umfangreicher Texte (Kategorie: Arbeitszeitkostenreduktion)		
Auftraggeber/in:	Enzo Hilzinger, Michael Lang		
Tragweite	Deutschland		
Projektleiter/in:	Michel Medved		
Projektzeitraum:	Geplanter Beginn: 22.05.2023 Geplantes Ende: 27.07.2023		9 Wochen
Erstellungsdatum:	16.05.2023	Budget: 0 €	

2.2 Beschreibung der Ausgangssituation

Die Mitarbeiter von Unternehmen und Organisationen, als auch private Einzelpersonen sehen sich in Zeiten exponentiell steigender Datenmengen zunehmend mit der Herausforderung konfrontiert, große Mengen an Informationen zu verarbeiten, um sie in ihrer Entscheidungsfindung miteinzubeziehen. Dabei werden verschiedene Informationsquellen genutzt (z.B. Forschungspapiere, Reports, Nachrichtenartikel, soziale Netzwerke). Generell ist der Verarbeitung von Informationen aus unstrukturierten Daten sehr zeitaufwändig, wobei Text den überwiegenden Teil unstrukturierter Daten ausmacht. Einzelpersonen haben dabei aufgrund des manuellen Lesens nur begrenzte Kapazitäten, um große Mengen an Text für ihre Entscheidungsfindung zu erfassen. Mit Hilfe eines Tools zur Textzusammenfassung und Textklassifikation könnte der Zeitaufwand für Einzelpersonen deutlich verringert werden, um die folgenden Zustände zu verbessern:

- Personen können aus Zeitgründen nur eine begrenzte Menge/Auswahl an Textinformationen erfassen und betrachten dadurch nur einen Bruchteil aller zugänglichen Informationen.
- Informationen werden je nach Leser unterschiedlich interpretiert, bewertet und klassifiziert, was zu Inkonsistenzen in der Entscheidungsfindung je nach Person/Leser führt.
- Verlangsamte Erfassung wichtiger Informationen führt zu verzögerter Reaktion auf Entwicklungen und Trends, wodurch für Unternehmen ein Wettbewerbsnachteil entsteht.

↳ nicht in Scope der Hilzinger

2.3 Definition der Projektziele

Das übergeordnete Gesamtprojektziel ist die Entwicklung eines zuverlässigen Tools zur Textzusammenfassung und -klassifikation, wobei das Maß der Textzusammenfassung unter Angabe einer Kompressionsrate vom Nutzer definiert werden kann. Die einzelnen Teilziele des Projektes werden nach Prioritäten (A-C) in **Muss-, Soll- und Kann-Ziele** unterteilt.

Priorität A: Muss-Ziele	
A1	Für das Tool muss ein Algorithmus entwickelt werden, der anhand eines Eingabetextes eine prägnante und präzise Zusammenfassung liefert.
A2	Das Tool muss in der Lage sein unter Angabe einer Kompressionsrate den Text in die entsprechende Länge zusammenzufassen (gemessen an der Wortanzahl), wobei eine Kompressionsrate von 80% eine Verringerung der Wortanzahl um 80% bedeutet.
A3	Für das Tool muss ein Modell trainiert werden, welches den Text basierend auf vordefinierten Kategorien oder thematischen Merkmalen zuverlässig klassifiziert (gemessen an Accuracy, Sensitivity, Specifity und F1-Score, Rouge) ✓
A4	Das Tool muss eine User-Interface besitzen, die es Nutzern ermöglicht Text und Kompressionsrate einzugeben und auf die Ergebnisse der Textzusammenfassung/-klassifikation zuzugreifen → <i>Accuracy</i>
Priorität B: Soll-Ziele	
B1	Das Tool soll zusammen mit Nutzer/Schulungsmaterialien geliefert werden, um Nutzern den direkten Umgang mit dem Tool zu erklären und vereinfachen
B2	Das Tool soll in der Lage sein PDF -Dateien als gängigstes Speichermedium für Textinformationen entgegenzunehmen, um den darin befindlichen Text zusammenzufassen und zu klassifizieren
B3	Das Tool soll durch ausführliche Dokumentation anderen Entwicklern die Möglichkeit bieten das Tool weiterzuentwickeln und zu warten.
Priorität C: Kann-Ziele	
C1	Das Tool kann mehrere verschiedene Eingabetexte gleichzeitig übergeben bekommen unter Angabe unterschiedlicher Kompressionsraten und liefert die Ergebnisse aller Eingabetexte
C2	Das Tool kann die Ergebnisse der Textzusammenfassung/-klassifikation in wenigen Sekunden dem Nutzer bereitstellen

2.4 Projektabgrenzung

2.4.1 In-Scope

Daten	<ul style="list-style-type: none">• Es werden anfangs ausschließlich englischsprachige Texte verwendet• Für die Texteingabe werden lediglich die Datenformate .pdf und .txt unterstützt sowie manuell eingegebener Text• Es wird lediglich von .pdf Dateien mit simplem Layout bei der Eingabe ausgegangen (.pdf Dateien mit mehreren Spalten werden beispielsweise ausgeschlossen) → 127?
Werkzeuge	<p>Für die Entwicklung des Modells für die Textklassifikation und der Textkomprimierung werden Python-Bibliotheken wie Huggingface, Tensorflow und Scikit-Learn verwendet.</p> <p>Für die Erstellung des UI wird Streamlit genutzt. Für die Codeverwaltung und die Projektorganisation wird GitHub verwendet.</p> <p>Für die zeitliche Veranschaulichung der Projektplanung wird das interne Kanbanboard von Github verwendet.</p>
Leistungen	<ul style="list-style-type: none">• Tool mit UI zur Textklassifizierung und (inhaltliche) Komprimierung• Funktion der manuellen Texteingabe• Anwendung der trainierten Modelle auf das manuell Eingebene in Form einer Pipeline
Zeit	Das Projekt ist zeitlich auf den 27.07.2023 terminiert

2.4.2 Out-of-Scope

Daten	<ul style="list-style-type: none">• Implementation weiterer Sprachen• Konvertierung anderer Textformate• Berücksichtigung qualitativ schlechterer Texte aus beispielsweise eingescannten Dokumenten oder nicht unterstützten Dateiformaten• Nutzung des Produktes in einem produktiven Umfeld
Leistungen	<ul style="list-style-type: none">• Kundensupport• Nutzerschulungen• Softwareupdatelieferungen über das Ende des Projektes hinaus• Schnittstellen für die Anbindung Software Dritter• Haftung für die Erzeugnisse des Produktes

2.5 Risikoanalyse

2.5.1 Risiken während des Projektes

Datenrisiko	Mangelnder Zugang zu annotierten Textdaten mit Klassifikationslabel für das Trainieren des Textklassifikationsmodells <u>Gegenmaßnahme</u> : Entwicklung eines Hilfstools, um annotierte Textdaten für die Klassifikationsmodelltraining zu generieren
Technisches Risiko	Unzureichende Leistung der Textzusammenfassung/-klassifikation aufgrund ? <u>Gegenmaßnahme</u> : Zeitliche Einplanung umfangreicher Tests und Optimierungsiterationen mit einer Vielzahl von Texten verschiedener Quellen ✓
	Mangelnde Skalierbarkeit des Systems bei der Verarbeitung großer Textmengen. <u>Gegenmaßnahme</u> : Nutzung von Cloud-basierten Ressourcen und Implementierung von skalierbaren Infrastrukturen zur Optimierung der Lastenverteilung → Provider?
Akzeptanz Risiko	Mangelnde Akzeptanz der Nutzer aufgrund komplexer Bedienbarkeit des Tools <u>Gegenmaßnahme</u> : Entwicklung eines intuitiven User-Interfaces und die Bereitstellung von Nutzer-/Schulungsunterlagen
Management Risiko	Mangelnde Ressourcenplanung (Zeit) und unzureichendes Know-How zur Umsetzung der Anforderungen <u>Gegenmaßnahme</u> : Erstellung eines detaillierten Projektablaufplans und klare Identifikation von erforderlichen Fähigkeiten und Kapazitäten im Team

2.5.2 Risiken bei Nichtdurchführung des Projektes

Bei Nichtdurchführung des Projektes ist der Bachelorabschluss der Projektbeteiligten stark gefährdet, wodurch dieses Projekt unter allen Umständen umgesetzt werden muss. Zusätzlich ergeben sich bei der Nichtdurchführung des Projektes verpasste Chancen für die Projektbeteiligten:

- Aneignung wichtiger Aspekte für die erfolgreiche Projektrealisierung
- Lernen und Weiterentwicklung wichtiger Fähigkeiten für IT-Produktentwicklung und NLP

u Noten?

2.6 Phasen und Projektabhängigkeiten

Die Projektumsetzung erfolgt über einen mehrstufigen Realisierungsansatz mit unterschiedlichen Abhängigkeiten. Das Projekt wird in die Phasen **(1) Data Selection und Preprocessing**, **(3) Modelltraining**, **(4) Testing** und **(5) Projektabschluss** unterteilt bei denen verschiedene Abhängigkeiten berücksichtigt werden müssen.

Phase 1: Data Selection
Die Qualität des Tools, insbesondere die Funktion der Textklassifikation hängt maßgeblich von der Verfügbarkeit geeigneter Datenquellen zur Trainings- und Testdatenbeschaffung ab. Während dieser Phase werden öffentlich zugängliche annotierte Textdaten beschafft. Gegebenenfalls müssen eigene Datensätze erstellt werden, um bestimmte Textklassifikationen zu trainieren.
Phase 2: Data Preprocessing/Cleaning
Die Nutzereingabe von Texten sowie die Textdaten für das Modelltraining müssen in ein einheitliches harmonisiertes Format überführt werden, um unerwünschte Artefakte, Rauschen, Inkonsistenzen und Fehler zu minimieren. Daher wird in dieser Phase eine Datenbereinigungs-/Datenvorverarbeitungspipeline entwickelt, um dem Modell den zu analysierenden Text in einem geeigneten Format zu übergeben.
Phase 3: Modelltraining
Das Modelltraining erfordert qualitativ hochwertige, vorverarbeitete Textdaten als Eingabe. Der Erfolg des Modelltrainings hängt davon ab, dass die Daten in einem einheitlichen geeigneten Format vorliegen. Das Modelltraining erfordert eine ausreichende Rechenleistung und Speicherressourcen, um die Algorithmen effizient auszuführen. Es ist wichtig sicherzustellen, dass die erforderlichen Ressourcen vorhanden sind, um das Modelltraining innerhalb der vorgegebenen Zeit- und Budgetgrenzen durchzuführen.
Phase 4: Testing/Validation
Das Testen/Validieren des Modells erfordert ein trainiertes Modell und geeignete Testdaten zur Evaluierung des Tools. In dieser Phase werden trainierte Modelle anhand ihrer Zusammenfassungsergebnisse (gemessen an Wortanzahl) und Klassifikationsergebnisse (gemessen an Accuracy, Sensitivity, Specifity und F1-Score).
Phase 5: Projektabschluss
In der Projektabschlussphase werden alle abzugebenden projektrelevanten Unterlagen vom Team überprüft und der Projektverlauf wird vom gesamten Team reflektiert und ein Projektabschlussbericht angefertigt.



2.7 Projektorganisation

Rolle/Rollen	Name	E-Mail	Bemerkung
Projektleiter	Michel Medved	michel.medved@gmail.com	-
Developer	Tobias Kister	tobias.kister@pm.me	-
Developer	Daniel Schmitz	dschmitz01@web.de	-
Developer	Jan Neifeld	J.Dario@gmx.de	-
Auftraggeber	Enzo Hilzinger	enzo.hilzinger@sap.com	Auftraggeber
Auftraggeber	Michael Lang	mi.lang@sap.com	Auftraggeber

→ Change Request Handling?

3 LASTENHEFT

3.1 Einleitung

3.1.1 Allgemeines

Das vorliegende Lastenheft dient als umfassender Leitfaden für die Umsetzung des Projektes „Textstrip2.0“ (Projektnummer: 001) zur Entwicklung eines Tools zur Textzusammenfassung und -klassifikation und definiert die Anforderungen, das Konzept und Rahmenbedingungen des Projektes zur Erreichung der vordefinierten Projektziele. Das Dokument ist in seiner gültigen Form für alle unten aufgeführten Projektbeteiligten im Github-Repository im Ordner “docs” zugänglich. Link: <https://github.com/orgs/StripTheText/repositories>

Rolle	Name	E-Mail	Bemerkungen
Projektleiter	Michel Medved	michel.medved@gmail.com	-
Developer	Tobias Kister	tobias.kister@pm.me	-
Developer	Daniel Schmitz	dschmitz01@web.de	-
Developer	Jan Neifeld	j.Dario@gmx.de	-
Auftraggeber	Enzo Hilzinger	enzo.hilzinger@sap.com	-
Auftraggeber	Michael Lang	mi.lang@sap.com	-

3.1.2 Meeting-Protokolle

Die Meetings finden sowohl gemeinsam mit den Auftraggebern statt als auch ausschließlich im Kreis des Projektteams, um bei Bedarf wichtige Entscheidungen wie auch allgemeine Anforderungen mit den Auftraggebern abzustimmen und gleichzeitig die technische Umsetzung im Projektteam vorzunehmen.

Datum	Thema	Teilnehmer	Zusammenfassung
08.05.2023	Projektauftrag & Projektteambildung	Projektteam + Auftraggeber	Empfang des Projektauftrags zur Textklassifikation- und Kompression, Projektteamzusammenstellung
14.05.2023	Brainstorming & Aufgabenverteilung	Projektteam	Brainstorming über Projekthinhalte und -ziele, Verteilung erster Zuständigkeiten
17.05.2023	Absprache	Projektteam	Diskussion der Anforderungen und Verteilung zugehöriger Aufgaben
17.05.2023	Austausch über Anforderungen	Projektteam + Auftraggeber	Absprache mit Stakeholdern über Anforderungen und Projektscope
21.05.2023	Dokumentation	Projektteam	Finalisierung der Dokumentation
22.05.2023	Termingestaltung & Anforderungen	Projektteam + Auftraggeber	Anpassung des Projektscope und Präzisierung der Anforderungen durch die Stakeholder
24.05.2023	Kick-Off	Projektteam + Auftraggeber	Vorstellung der Ziele und Anforderungen, Beginn der Durchführungsphase

3.2 Konzept und Rahmenbedingungen

wie umfangreich?
↑

3.2.1 Benutzer / Zielgruppe

Das Studienprojekt richtet sich an eine breite Online-Zielgruppe, die regelmäßig umfangreiche Texte recherchiert und dabei den Zeitaufwand reduzieren möchte. Dazu gehören sowohl Studierende aller Semester, die für ihre akademischen Arbeiten relevante Informationen aus einer Vielzahl von Quellen sammeln müssen, als auch professionelle Forscher und Wissenschaftler, die regelmäßig umfangreiche literarische Werke, wissenschaftliche Artikel oder Dokumentationen durchgehen. Neben einer wissenschaftlich motivierten Zielgruppe zielt das Projekt auch darauf ab, Menschen aus anderen Bereichen wie dem Journalismus und Blogger dabei zu helfen, relevante Informationen schneller zu identifizieren und daraus eine kompakte Zusammenfassung zu generieren, die den Zeitaufwand erheblich verringert. Zudem sind auch Käuferinnen und Käufer eine Zielgruppe, die durch die Möglichkeit der Zusammenfassung von Produktbewertungen adressiert werden sollen.

3.2.2 Ziele und Nutzen für den Anwender

Die Ziele und Nutzen für den Anwender werden als User-Stories dokumentiert:

1. Als Nutzer möchte ich in der Lage sein, einen umfangreichen Text über das Tool hochzuladen und anschließend eine Zusammenfassung der wichtigsten Informationen zu erhalten.
2. Als Nutzer möchte ich bestimmen, mit welcher prozentualen Kompressionsrate dieser Text zusammengefasst.
3. Als Nutzer möchte ich einen dem System unbekannten Text hochladen können und diesen durch ein KI-Modell in eine Textart klassifizieren lassen.
4. Als Nutzer möchte ich in der Lage sein, die Anwendung über ein bedienfreundliches Frontend zu nutzen.
5. Als Nutzer benötige ich zur erfolgreichen Verwendung des Tools geeignete Nutzer-/Schulungsunterlagen.
6. Als Entwickler möchte ich den zu Grunde liegenden Code verstehen und die Möglichkeit besitzen das Tool zu warten und weiterzuentwickeln.

→ gute Darstellung

3.2.3 Systemvoraussetzungen und Ressourcen

Für ein Minimum Viable Product (MVP) zur Textklassifizierung und Textkompression sind bestimmte Systemvoraussetzungen erforderlich. Insbesondere für das Training der Modelle wird mindestens 16 GB RAM empfohlen, um ausreichend Speicher für die Datenverarbeitung und Datenhaltung zum Modelltraining bereitzustellen. Optimalerweise sollten jedoch 32 GB RAM vorhanden sein, um das Ausbleiben von Out-Of-Memory-Fehlern während des Trainings zu gewährleisten. Darüber hinaus kann die Nutzung von einer GPU das Training erheblich beschleunigen, da GPUs effizienter bei der parallelen Berechnung zum Training neuronaler Netze sind. Diese Systemanforderungen tragen dazu bei, dass das MVP effektiv und effizient arbeitet, indem genügend Ressourcen für das Training und die Verarbeitung der Textdaten bereitgestellt werden.

→ welche Hardwareanforderungen?

3.3 Anforderungsbeschreibung

3.3.1 Funktionale Anforderungen

Anforderung: Benutzeroberfläche [FA_001]	
Beschreibung	Das Tool soll eine benutzerfreundliches User-Interface haben, das intuitiv zu bedienen ist. Das Design sollte ansprechend und responsiv sein. Das UI sollte responsive sein und auf verschiedenen Geräten und Bildschirmauflösungen einwandfrei funktionieren. Es sollte auch eine einheitliche Benutzererfahrung auf verschiedenen Browsern bieten. ✓
Wechselwirkung	Wechselwirkungen gibt es mit allen vorgesehenen Funktionen des Webtools, da diese auf dem User-Interface integriert werden müssen. Funktionen mit Wechselwirkung: <ul style="list-style-type: none">• Eingabefeld(er)• Textkomprimierung• Textklassifizierung <i>3 UI</i> <i>Accessibility?</i>
Risiken	<ul style="list-style-type: none">• Mangelnde Akzeptanz der Benutzer bei fehlender Intuitivität• Kompatibilitätsprobleme durch verschiedene Geräte• Schulungsaufwand bei zu schwerer Bedienbarkeit• Produktivitätsverlust bei fehlender Intuitivität
Aufwandsschätzung	2 Personentage

Anforderung: Texteingabe [FA_002]	
Beschreibung	Die Eingabe eines Textes soll nutzerseitig möglich sein, inklusive nachgelagerter Verarbeitung des Inputs durch die gewünschte Funktionalität. Das Eingabefeld sollte intuitiv nutzbar und für den Nutzer einfach zu finden sein, um die Bedienbarkeit optimal zu gestalten. Der Text soll via einfache Texteingabe über eine Konsole oder via PDF-Datei vom Nutzer übergeben werden können. <i>txt?</i>
Wechselwirkung	Die Funktion der Texteingabe muss verbunden werden mit den Funktionen der Textzusammenfassung und -klassifikation und muss eine Schnittstelle zur Vorverarbeitungspipeline liefern. Konkret muss der Inhalt der Eingabe korrekt an die Funktionen übergeben werden.
Risiken	<ul style="list-style-type: none">• Einschränkung der Funktionalitäten der Textkomprimierung und -klassifikation, da keine Nutzereingaben verwertet werden können• Fehlerhafte Ausgaben der nachfolgenden Funktionen aufgrund von fehlerhafter Datenübertragung• Geringe Nutzungseffizienz verursacht durch fehlende Nutzerfreundlichkeit im Aufbau
Aufwandsschätzung	2 Personentage

Anforderung: Ergebnis-Ausgabe [FA_003]	
Beschreibung	Die zu entwickelnde Komponente dient zur Darstellung der Ergebnisse von Textzusammenfassungen und Textklassifizierungen. Sie muss in der Lage sein, komplexe Datenstrukturen von Zusammenfassungen und Klassifizierungen in einer klaren, leicht verständlichen und übersichtlichen Form darzustellen. Das können grafische Darstellungen, Tabellen oder Listen sein. Sie soll Benutzern ermöglichen, die Ergebnisse effektiv zu analysieren und zu interpretieren.
Wechselwirkung	<p>Die Komponente interagiert mit dem Textzusammenfassungs- und Textklassifizierungs-Modul, indem sie dessen Ausgabedaten empfängt und verarbeitet. Sie muss in der Lage sein, große Mengen von Daten effizient zu verarbeiten und darzustellen.</p> <p>Außerdem interagiert sie mit dem Benutzer-Interface. Dieses stellt die von der Komponente erzeugten Ausgaben dar und ermöglicht es dem Benutzer, mit den Ergebnissen zu interagieren. Daher sollte die Komponente in der Lage sein, Daten in einem Format zu liefern, das vom Interface leicht interpretiert und dargestellt werden kann.</p>
Risiken	<p>Ein Risiko könnte in der Performance liegen. Bei der Verarbeitung großer Datenmengen könnte die Komponente ineffizient werden, was zu langsamen Reaktionszeiten oder gar Systemabstürzen führen kann.</p> <p>Ein weiteres Risiko könnte in der Qualität der Datenvisualisierung liegen. Wenn die Darstellung der Ergebnisse für den Benutzer nicht klar und verständlich ist, könnte dies zu Missverständnissen und falschen Interpretationen führen. Schließlich könnte die Weiterverarbeitung der Ergebnisse ein Risiko darstellen. Wenn die Exportfunktionen nicht korrekt implementiert sind, könnten die Daten in einem unbrauchbaren Format ausgegeben werden, was die Weiterverarbeitung der Ergebnisse erschwert. Um diese Risiken zu mindern, sollte das Entwicklungsteam Best Practices für Datenverarbeitung und Visualisierung anwenden und regelmäßige Tests durchführen, um die Leistung und Funktionalität der Komponente sicherzustellen.</p>
Aufwandsschätzung	1 Personentag

Anforderung: Textzusammenfassung [FA_004]	
Beschreibung	<p>Das Softwareprodukt soll über eine Funktion verfügen, die in der Lage ist, englischsprachige Texte zu komprimieren. Die <u>Kompressionsrate</u> soll dabei vom Endnutzer definiert werden können. Der komprimierte Text soll trotz der Reduzierung der Wortanzahl korrekte Rechtschreibung und Grammatik aufweisen. Diese Funktion ist wichtig, um den Endnutzern eine effiziente Möglichkeit zu bieten, umfangreiche Texte zu vereinfachen und zu verkürzen, ohne dabei wichtige Informationen oder den Sinn des Textes zu verlieren. <i>→ Kryptisch hier?</i></p>
Wechselwirkung	<p>Interaktion mit der Benutzererfahrung: Die Textkomprimierung kann die Benutzererfahrung beeinflussen, abhängig von ihrer Anpassbarkeit und der Präferenz des Benutzers.</p> <p>Interaktion mit Leistung und Geschwindigkeit: Abhängig von Textumfang und Komplexität des Zusammenfassungsmodell könnte die Textkomprimierung die Systemleistung beeinflussen.</p>
Risiken	<p>Verlust von Informationen: Es besteht das Risiko, dass wichtige Informationen bei der Komprimierung verloren gehen, insbesondere bei einer hohen Kompressionsrate. Es muss sichergestellt werden, dass die Software sorgfältig entwickelt wird, um das Risiko eines Informationsverlustes zu minimieren.</p> <p>Komplexität der Sprachverarbeitung: Die korrekte Berücksichtigung von Grammatik und Rechtschreibung bei der Komprimierung kann technisch herausfordernd sein. Ein Risiko besteht darin, dass trotz sorgfältiger Programmierung Fehler auftreten können.</p> <p>Zeit und Ressourcen: Die Implementierung einer solchen Funktion kann zeitaufwendig sein und erhebliche Ressourcen erfordern. Es besteht das Risiko, dass die Entwicklung dieser Funktion andere Aspekte des Projekts verzögert oder die verfügbaren Ressourcen übersteigt.</p>
Aufwandsschätzung	8 Personentage (Primäre Priorität)

% der Anzahl?

Anforderung: Textklassifikation [FA_005]	
Beschreibung	Mit einer Genauigkeit von mindestens <u>70%</u> soll das Softwareprodukt Texte anhand vorgegebener Kategorien in verschiedene Textarten klassifizieren. Dazu soll ein <u>ML</u> -Modell trainiert werden, das über annotierte Trainingsdaten eine Zuordnung zwischen Textcharakteristiken und Textklassen annähert.
Wechselwirkung	Die Umsetzung der Textklassifizierung hängt stark von dem Preprocessing und die dazu gestellten Anforderungen ab. Bei Veränderungen der Preprocessing-Pipeline wird auch die Textklassifizierungsgenauigkeit beeinflusst. Zudem ist die Textklassifizierung auch von der Anforderung zur Texteingabe und den verschiedenen Eingabeformaten abhängig.
Risiken	Die verwendete Datengrundlage ist unzureichend oder die angewendeten Algorithmen können keine eindeutigen Muster identifizieren, sodass die Textklassifizierungsgenauigkeit nicht die geforderten <u>80%</u> erfüllt. Bei zu vielen Textkategorien oder Kategorien mit ähnlichen Charakteristiken kann ebenfalls die Klassifizierungsgenauigkeit geringer ausfallen. Zudem sollte auch ein Stichtag definiert werden, um nicht durch die Aufnahme weitere Klassen die Datengrundlage bzw. Preprocessing-Pipeline anpassen zu müssen und ein neues Modell zu trainieren.
Aufwandsschätzung	6 Personentage

80 - d.h.
70?

Anforderung: Backend [FA_006]	
Beschreibung	Das Backend des Softwareprodukts soll die Anfragen asynchron verarbeiten, damit eine Verarbeitung der Eingabe nicht die gesamte Applikation blockiert. Zudem soll die Implementierung in <u>Python</u> erfolgen und es muss eine Funktion im Backend vorhanden sein, um Dateiuploads zwischenspeichern.
Wechselwirkung	Zum einen wirkt sich die Gestaltung des Backends auf die Anforderung der Benutzeroberfläche aus, da es aus der Benutzeroberfläche aufgerufen werden muss. Zum anderen ist das Backend von den Modellen der Textkomprimierung und Textklassifizierung abhängig. Es muss diese Modelle mit den geforderten Eingabewerten aufrufen können und über genügend Ressourcen verfügen.
Risiken	Ein schlecht strukturiertes oder unzureichend dokumentiertes Backend kann zu erhöhtem Aufwand bei der Wartung und Weiterentwicklung führen. Änderungen oder Fehlerbehebungen können schwierig sein, insbesondere da die Logik von dem Frontend abstrahiert wurde.
Aufwandsschätzung	4 Personentage

Anforderung: PDF-Textextraktion [FA_007]	
Beschreibung	Das Tool soll in der Lage sein neben der einfachen Texteingabe auch PDF-Dateien als Texteingabe zu verwenden. Dafür muss der Text zunächst aus dem Dokument extrahiert und in ein geeignetes Format überführt werden, um den PDF-Text nach Vorverarbeitung dem Modell zu übergeben. Zusätzlich soll dieses Hilfstool gegebenenfalls zur Erstellung eigener annotierter Datensätze für das Modelltraining verwendet werden.
Wechselwirkung	Das Hilfstool soll gegebenenfalls der Erstellung eigener annotierter Datensätze dienen und beeinflusst damit die Datengrundlage auf dem das Modell trainiert wird. Zusätzlich soll dieses Tool auch dafür genutzt werden, um dem Nutzer die Texteingabe via PDF-Dateien als gängigstes Dateiformat für Texte zu ermöglichen.
Risiken	Bei der Nutzung von PDF-Dateien besteht die Möglichkeit, dass qualitativ unzureichende Dateien für das Tool verwendet werden (z.B. schlechte/unlesbare PDF-Scans), die die Textextraktion und damit die Ergebnisse des Tools negativ beeinflussen. ✓
Aufwandsschätzung	5 Personentage

Anforderung: Preprocessing [FA_008]	
Beschreibung	Innerhalb der Toolchain wird eine Vorverarbeitungspipeline benötigt, um dem Modell immer ein qualitativ einheitliches Datenformat in benötigter Form zu übergeben. Die Vorverarbeitung von Texten für NLP-Zwecke umfasst die Tokenisierung, Groß-Kleinschreibungskorrektur, Entfernung von Satzenden, Stemming und die Lemmatisierung.
Wechselwirkung	Die Vorverarbeitung der Eingabetexte ist essenziell und hat direkten Einfluss auf das Modell und dessen Performance.
Risiken	Unzureichende Vorverarbeitung der Texte sorgt für uneinheitliche und fehlerhafte Eingabe von Textinformationen, was die Performance des trainierten Modells erheblich beeinflusst und zu einem schlechten Endprodukt führen kann.
Aufwandsschätzung	4 Personentage

3.3.2 Nicht funktionale Anforderungen

Anforderung: Wartbarkeit [NFA_001]	
Beschreibung	Das Tool soll für andere Entwickler optimierbar und wartbar sein, um das Tool für entsprechende Zwecke weiterzuentwickeln. <i>→ wie?</i>
Wechselwirkung	Die Wartbarkeit des Tools hat direkten Einfluss auf die Zufriedenheit und Akzeptanz der Auftraggeber gegenüber dem Tool.
Risiken	Unzureichende Code-Dokumentation können dazu führen, dass das Tool von den Entwicklern nicht verwendet wird und damit auch nicht weiter optimiert und weiterentwickelt wird.

Anforderung: Nutzer-/Schulungsmaterialien [NFA_002]	
Beschreibung	Um den Nutzer einen einfachen Einstieg in die Verwendung des Tools zu ermöglichen werden entsprechende Nutzerunterlagen benötigt. Diese Unterlagen umfassen unter anderem: Installationshinweise, Anleitung zur Verwendung und Übersicht der verwendbaren Funktionen. <i>→ welche Form?</i>
Wechselwirkung	Die Nutzer-/Schulungsmaterialien beeinflussen die Akzeptanz des Tools für den Endnutzer definieren den Projekterfolg gemessen an effektiver Nutzeranzahl.
Risiken	Unzureichende Nutzer-/Schulungsmaterialien können dazu führen, dass das Tool von den Nutzern nicht verwendet wird und damit auch keinen Mehrwert liefert.

3.3.3 Abnahmekriterien

Die folgenden Abnahmekriterien dienen als Kontrollinstrumente zum Projektabschluss und ermöglichen eine Bewertung des Projekterfolgs:

- Die Bedienbarkeit der Benutzeroberfläche kann durch eine Prüfung aller wesentlichen bereitgestellten Funktionen evaluiert werden.
- Die Funktionalität der manuellen Texteingabe kann mithilfe verschiedener Testfälle, wie beispielsweise variierenden Textlängen, geprüft werden.
- Die Funktionalität der Textklassifikation kann oberflächlich über ein manuell gelabeltes Testdatenset geprüft werden, genauer durch den Vergleich mit bereits etablierten Tools.
- Das Maß der Textzusammenfassung darf nur geringe Abweichungen bei der Textkomprimierung liefern gemessen an der Wortanzahl des Eingabetextes
- Die sinngemäße Textzusammenfassung kann nur oberflächlich durch manuelles Lesen geprüft werden inklusive Rechtschreibung und Grammatik geprüft. *→ A/B-Test*
- Die intuitive Bedienbarkeit der Benutzeroberfläche kann lediglich subjektiv durch eine Testung verschiedener Nutzer evaluiert werden. *→ User Research (usability)*
- Das Tool wird zusammen mit Nutzerunterlagen und einer detaillierten Dokumentation geliefert

4 PFLICHTENHEFT

4.1 Einleitung

4.1.1 Allgemeines

Das vorliegende Pflichtenheft dient als umfassender Leitfaden für die Umsetzung der einzelnen Anforderungen, die im entsprechenden Lastenheft zum Projekt „**Textstrip2.0**“ (Projektnummer: **001**) zur Entwicklung eines Tools zur Textzusammenfassung und -klassifikation und definiert wurden. Das Dokument ist in seiner gültigen Form für alle unten aufgeführten Projektbeteiligten im Github-Repository im Ordner “docs” zugänglich.

Link: <https://github.com/orgs/StripTheText/repositories>

Rolle	Name	E-Mail	Bemerkungen
Projektleiter	Michel Medved	michel.medved@gmail.com	-
Developer	Tobias Kister	tobias.kister@pm.me	-
Developer	Daniel Schmitz	dschmitz01@web.de	-
Developer	Jan Neifeld	j.Dario@gmx.de	-
Auftraggeber	Enzo Hilzinger	enzo.hilzinger@sap.com	-
Auftraggeber	Michael Lang	mi.lang@sap.com	-

4.1.2 Meeting-Protokolle

Siehe 3.1.2 Lastenheft

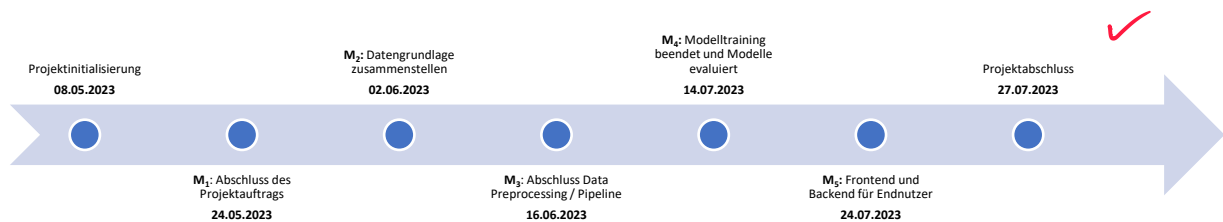
4.2 Ressourcen

Innerhalb der Ressourcenplanung sind vier Projektteammitglieder mit je 15 **PT** vorgesehen. Da das Budget 0€ umfasst, soll bei Hardware und Cloudservices auf bereits laufende oder kostenlose Dienstleistungen zurückgegriffen werden. Gleichmaßen soll auch auf Open Source-Software zurückgegriffen werden, um keine Kosten bei der Softwarewahl zu verursachen.

Ressourcenplanung		
Kategorie	Geplanter Ressourcenbedarf	Tatsächlicher Ressourcenbedarf
Menschen	4 Personen á 15 PT (Σ 60 PT)	-
Budget	0€	-
Hardware & Cloudservices	50h x mindestens 16 GB RAM zum Modelltraining Platform-as-a-Service je für Frontend & Backend	-
Lizenzen	Nutzung von Open Source-Software	-

4.3 Meilensteine

Als Meilensteine können fünf Zeitpunkte im Projektverlauf definiert werden. Der erste Meilenstein M_1 wird mit der Kick-Off-Präsentation und Abschluss des Projektauftrags sowie der initialen Fassung des Lasten- und Pflichtenhefts am 24.05.2023 erreicht. Der zweite Meilenstein M_2 wird mit der Zusammenstellung einer geeigneten Datengrundlage erreicht und ist planmäßig auf den 02.06.2023 terminiert. Mit zwei weiteren Wochen ist das Erreichen des dritten Meilensteins M_3 für den 16.06.2023 geplant, indem eine erste Pipeline zum Preprocessing der Daten erstellt wird. Im Anschluss dessen kann das Modelltraining erfolgen. Als Meilenstein M_4 soll bis zum 14.07.2023 die trainierten Modelle evaluiert werden. Abschließend soll bis zum 24.07.2023 das Frontend wie auch das Backend für die Endnutzerinnen und Endnutzer entwickelt werden. *gute Struktur*



4.4 Umsetzung der vordefinierten Anforderungen

4.4.1 Funktionale Anforderungen

Umsetzung der Anforderung: Benutzeroberfläche [FA_001]	
Beschreibung	<p>Das Tool soll eine benutzerfreundliches User-Interface haben, das intuitiv mit einem ansprechenden Design zu bedienen ist. Das UI sollte responsive sein und auf verschiedenen Geräten und Bildschirmauflösungen funktionieren.</p> <p>Es sollte auch eine einheitliche Benutzererfahrung auf verschiedenen Browsern bieten. Dazu soll das Frontend mithilfe von Streamlit aufgebaut werden, wodurch grundlegende Funktionalitäten vereinfacht umgesetzt werden können.</p>
Wechselwirkung	<p>Wechselwirkungen können mit allen implementierten Funktionen auftreten, darunter die Textklassifikation, Textzusammenfassung, Ergebnisdarstellung und die Eingabemöglichkeit von Text.</p> <p>Probleme können dabei durch unterschiedliche Versionierungen der verwendeten Pakete auftreten</p>
Testhinweise	<p>Die Funktionalität aller Elemente muss manuell in verschiedenen Szenarien getestet werden, dazu müssen konkrete Testfälle definiert werden, wie die zu testenden Browser und Endgeräte.</p> <p>Beispiele wären Kombinationen verschiedener Auflösungen und Seitenverhältnisse in unterschiedlichen Browsern. <i>→ welche?</i></p>
Aufwandsschätzung	4 Personentage

Umsetzung der Anforderung: Texteingabe [FA_002]	
Beschreibung	Die Eingabe eines Textes soll nutzerseitig möglich sein, inklusive nachgelagerter Verarbeitung des Inputs durch die gewünschte Funktionalität. Das Eingabefeld sollte intuitiv nutzbar und für den Nutzer einfach zu finden sein, um die Bedienbarkeit optimal zu gestalten. Der Text soll via einfache Texteingabe über eine Konsole oder via PDF-Datei vom Nutzer übergeben werden können.
Wechselwirkung	Wechselwirkungen können durch verschiedene Versionierungen der benutzten Pakete auftreten. Durch abweichende Eingabeformate können ebenfalls Fehler auftreten.
Testhinweise	Zur Fehlervermeidung kann ein möglichst breites Testfeld mit verschiedenen formatierten Eingabetexten und PDF-Dateien unterschiedlicher Qualität definiert werden.
Aufwandsschätzung	2 Personentage

Umsetzung der Anforderung: Ergebnis-Ausgabe [FA_003]	
Beschreibung	Die Komponente wird entwickelt, um die Ergebnisse von Textzusammenfassungen und Textklassifizierungen darzustellen. Sie wird dazu programmiert, komplexe Datenstrukturen in einer klaren, leicht verständlichen und übersichtlichen Form darzustellen, z.B. in Form von grafischen Darstellungen, Tabellen und Listen. <i>→ original?</i>
Wechselwirkung	Die Komponente interagiert mit dem Textzusammenfassungs- und Textklassifizierungs-Modul sowie mit dem Benutzer-Interface. Sie nimmt die Ausgabedaten auf, verarbeitet diese und stellt sie dar. Die Komponente muss daher in der Lage sein, Daten in einem Format zu liefern, das vom Interface leicht interpretiert und dargestellt werden kann.
Testhinweise	Um die Risiken zu mindern, wird das Entwicklungsteam Best Practices für Datenverarbeitung und Visualisierung anwenden und regelmäßige Tests durchführen. Es wird sowohl auf die Performance (Verarbeitungsgeschwindigkeit und Systemstabilität bei großen Datenmengen) als auch auf die Qualität der Datenvisualisierung (Klarheit, Verständlichkeit) und Datenexportfunktionen geachtet.
Aufwandsschätzung	1 Personentag

Umsetzung der Anforderung: Textzusammenfassung [FA_004]	
Beschreibung	Die zu entwickelnde Komponente wird zur Textzusammenfassung eingesetzt. Sie verwendet ein Transformer-Modell, das mithilfe von Transfer Learning vortrainiert wurde (mit Modellen von Huggingface). Die Eingabe in das Modell stellt ein durch das Preprocessing aufbereiteter Text dar. Mögliche Ausgabeoptionen sind Stichpunkte und zusammengefasster Text. Die Kompressionsrate der Texte kann über das Frontend eingestellt werden, z.B. 80% bedeutet, dass der resultierende Text 80% weniger Wörter als der Originaltext enthalten soll.
Wechselwirkung	Die Komponente interagiert mit verschiedenen Modulen. Sie erhält Eingaben und leitet sie an die Preprocessing-Pipeline weiter. Danach speichert sie die verarbeiteten Daten in einem Datei-Speicher. Sie erzeugt eine Ausgabe, die von der Ergebnis-Ausgabe-Komponente dargestellt wird, und kommuniziert mit dem Frontend zur Einstellung von Parametern, einschließlich der Kompressionsrate.
Testhinweise	Die Tests für diese Komponente umfassen die Verwendung der Rouge-Metrik zur Bewertung der Textzusammenfassungen, manuelle Überprüfungen und Cross-Validation-Verfahren zur Sicherstellung der Qualität und Genauigkeit der Ausgabe.
Aufwandsschätzung	8 Personentage

Umsetzung der Anforderung: Textklassifikation [FA_005]	
Beschreibung	<p>Zur Textklassifikation wird entweder über Transfer Learning ein bestehendes ML-Modell, beispielsweise auf Basis von Transformers, angelernt oder ein eigenständiges Modell in Form eines rekurrenten neuronalen Netzwerks trainiert, das die Klassen Nachrichtenartikel, Bücher, Blog-Beiträge, Produkt-Reviews und wissenschaftliche Veröffentlichungen mit einer Genauigkeit von mindestens 70% identifizieren soll. Hierbei können in Abhängigkeit der zur Verfügung stehenden Datengrundlage und verändernder Anforderungen weitere Klassen hinzugefügt werden, wenngleich die maximale Anzahl an Klassen aus Komplexitätsgründen nicht acht übersteigen darf. Über einen bereits vorverarbeiteten Eingabetext mit maximal <u>1500 Wörtern</u> soll das ML-Modell eine Klassifizierung vornehmen und als Ausgabe einen Vektor mit entsprechenden Wahrscheinlichkeiten für die einzelnen Klassen je in dem Intervall [0;1] zurückgeben.</p>
Wechselwirkung	<p>Durch die Übergabe eines Eingabetexts entsprechend der Eingabe nach Anforderung FA_002 und der vorgelagerten Preprocessing-Pipeline nach Anforderung FA_008 besteht eine Abhängigkeit zu den Umsetzungen dieser Anforderungen. Darüber hinaus sind auch nachgelagerte Aktivitäten wie das Backend nach Anforderung FA_006 und die Ergebnisausgabe nach Anforderung FA_003 abhängig. Hierbei soll über das Backend das Textklassifikationsmodell weitestgehend von der Ergebnisausgabe entkoppelt werden.</p>
Testhinweise	<p>Die Qualität der Textklassifikation kann anhand üblicher Metriken für Klassifizierungsprobleme wie der Accuracy, der Precision, dem Recall und F1-Score bestimmt werden. Hierbei ist zu betonen, dass zum Testen eine zu den Trainingsdaten disjunkte Datenmenge verwendet wird. <i>recall? class-Verbindlichkeit?</i></p>
Aufwandsschätzung	6 Personentage

Umsetzung der Anforderung: Backend [FA_006]	
Beschreibung	<p>Das Backend dient dazu, die Modelle zur Textklassifikation und Textkompression aufzurufen und deren Ergebnisse bereitzustellen. Dabei soll für die Textkompression die Textklasse als weiteres Feature neben der vorverarbeiteten Texteingabe beigelegt werden. Das Schnittstellenformat des Backend ist <u>JSON</u>, um die Integration in verschiedene Systeme zu gewährleisten. Das konkrete Framework wird auf Basis technischer Eigenschaften im Verlauf des Projekts ausgewählt. <i>-> abh. Python</i></p>
Wechselwirkung	<p>Zum einen ist die Ergebnisausgabe nach Anforderung FA_003 und damit auch die Benutzeroberfläche nach Anforderung FA_001 von dem Backend abhängig. Über die Definition eines JSON-Formats im Projektverlauf wird diese Abhängigkeit zwar beibehalten, jedoch auch das Schnittstellenformat transparent gemacht. Durch die Entkopplung der Modelle ist das Backend wiederum von den Anforderungen FA_004 bzw. FA_005 und deren Umsetzung abhängig. Dabei steht nicht nur die Modelleingabe, sondern auch die Modellausgabe in einer Wechselwirkung. Um dieser entgegenzuwirken, wird auch hier das Schnittstellenformat im Projektverlauf transparent gemacht. ✓</p>
Testhinweise	<p>Die korrekte Funktionalität des Backends und Erfüllung der Anforderungen soll über Unit-Tests geprüft werden. Zusätzlich soll über manuelle Tests die Integration in das Frontend geprüft werden. Testgegenstände sind hierbei:</p> <ul style="list-style-type: none"> • Der Aufruf des Textklassifikations- bzw. Textkompressionsmodells • Die Ergebnissrückgabe der jeweiligen Modelle • Die korrekte Zwischenspeicherung von PDF-Dateien zur Textextraktion
Aufwandsschätzung	4 Personentage

Umsetzung der Anforderung: PDF-Textextraktion [FA_007]

Beschreibung	<p>Die PDF-Textextraktion soll es dem Nutzer ermöglichen die Textzusammenfassung und -klassifikation auf Basis einer PDF-Dateien zu generieren. Außerdem soll das Tool gegebenenfalls zur Erstellung eigener annotierter Datensätze verwendet werden. Zur Extraktion von Text wird eine <u>OCR</u>-Pipeline benötigt, die den Prozess der Textextraktion in mehrere Schritte unterteilt:</p> <ol style="list-style-type: none"> 1) Einlesen der PDF 2) Segmentierung der Seiten und Erstellung von Bilddateien 3) Textextraktion von Bildern 4) Speichern der extrahierten Datei in eine txt-Datei <p>Für die OCR-Textextraktion wird das pyTesseract als Open Source Tool verwendet. Der Prozess der Textgenerierung soll vollkommen automatisiert werden, so dass ein Eingabeordner mit darin befindlichen PDF-Dateien die korrespondierenden txt-Dateien liefert.</p>
Wechselwirkung	<p>Das PDF-Extraktionstool kann den Projektverlauf früh positiv beeinflussen, da es die Möglichkeit liefert weitere annotierte Textdaten für das Modelltraining zu generieren. Außerdem beeinflusst es die User-Experience des Tools, weil der Nutzer die Möglichkeit hat Text via PDF-Datei dem Tool zu übergeben.</p>
Testhinweise	<p>Die OCR-Textextraktion kann nur überwiegend stichprobenartig manuell kontrolliert werden, wobei auf folgende Punkte zu achten ist:</p> <ul style="list-style-type: none"> • Texte sind unterschiedlich formatiert, z.B. Forschungspapiere überwiegend in einem "Two-Column Textformat". Dadurch muss auf die richtige Reihenfolge bzw. Erkennung des Formats geachtet werden, so dass der Text auch in der richtigen Reihenfolge extrahiert wird. • Es sollte ebenfalls geprüft werden wie sich die OCR-Textextraktion bei Abbildungen und Tabellen verhält <i>+ Dokumentation</i> • Die Überprüfung der extrahierten Texte kann über einen visuellen Ansatz unter Erstellung von "bounding box images" generiert werden, um zu prüfen welche Worte/Abschnitte nicht vom OCR-System erkannt werden
Aufwandsschätzung	4 Personentage

Umsetzung der Anforderung: Preprocessing Pipeline [FA_008]	
Beschreibung	<p>Eine Vorverarbeitungspipeline wird benötigt, um dem Modell immer ein standardisiertes Datenformat zu übergeben und die Datenqualität der Eingabetexte zu verbessern.</p> <ul style="list-style-type: none"> • Tokenisierung • Groß-Kleinschreibungskorrektur • Entfernung von Stopp-Worten • Stemming und Lemmatisierung ✓ <p>Für die Umsetzung einzelner Vorverarbeitungsschritte wird die NLTK-Bibliothek verwendet. Abhängig von bestimmten Textklassen müssen ggf. eigene Funktionen selbst implementiert werden.</p>
Wechselwirkung	Die Vorverarbeitung der Eingabetexte ist essenziell und hat direkten Einfluss auf das Modell und dessen Performance.
Testhinweise	Die Vorverarbeitungsschritte werden anhand kleiner manuell erstellter Textdaten überprüft, wobei geprüft wird ob die Vorverarbeitungsschritte wie erwartet ausgeführt werden.
Aufwandsschätzung	4 Personentage

4.4.2 Nicht funktionale Anforderungen

Anforderung: Wartbarkeit [NFA_001]	
Beschreibung	Das Tool soll für andere Entwickler optimierbar und wartbar sein, um das Tool für entsprechende Zwecke weiterzuentwickeln. Dazu soll der Code ausführlich mit Kommentaren und einer Dokumentation angereichert werden, um den Aufbau nachvollziehbar zu gestalten. <i>→ Code Qualität? Metriken?</i>
Wechselwirkung	Die Wartbarkeit des Tools hat direkten Einfluss auf die Zufriedenheit und Akzeptanz der Auftraggeber gegenüber dem Tool.
Risiken	Unzureichende Code-Dokumentation können dazu führen, dass das Tool von den Entwicklern nicht verwendet wird und damit auch nicht weiter optimiert und weiterentwickelt wird.

Anforderung: Nutzer-/Schulungsmaterialien [NFA_002]	
Beschreibung	Um den Nutzer einen einfachen Einstieg in die Verwendung des Tools zu ermöglichen werden entsprechende Nutzerunterlagen benötigt. Diese Unterlagen umfassen unter anderem: Installationshinweise, Anleitung zur Verwendung und Übersicht der verwendbaren Funktionen.
Wechselwirkung	Die Nutzer-/Schulungsmaterialien beeinflussen die Akzeptanz des Tools für den Endnutzer definieren den Projekterfolg gemessen an effektiver Nutzeranzahl.
Risiken	Unzureichende Nutzer-/Schulungsmaterialien können dazu führen, dass das Tool von den Nutzern nicht oder ineffizient verwendet wird und damit auch wenig bis keinen Mehrwert liefert.