SynTex

Texte klassifizieren und zusammenfassen

Dokumentation des Moduls "Projektrealisierung"

im Bereich Wirtschaftsinformatik: Data Science
an der Dualen Hochschule Baden-Württemberg Mannheim

Autoren:	Niklas Koch, Niclas Cramer, Jasmina Pascanovic & Antoine Fuchs
Matrikelnummern:	6699912, 7607733, 5711726 & 3008441
Kurs:	WWI20DSA
Studiengangsleiter:	Prof. Dennis Pfisterer
Dozent:	Michael Lang, Enzo Hilzinger
Abgabedatum:	24.05.2023

Inhaltsverzeichnis

1 Projektauftrag und -scope

Projekttitel	SynTex
Projektart	Studienprojekt
Projektteam	Niklas Koch, Jasmina Pascanovic, Antoine Fuchs, Niclas Cramer
Projektleitung	Jasmina Pascanovic
Kontaktdaten des Projektleiters	s201668@student.dhbw-mannheim.de
Projektauftraggeber	Enzo Hilzinger, Michael Lang
Projektdauer	08.05.2023 - 27.07.2023

Tabelle 1.1: Projektauftrag - Teil 1.

Ausgangssituation / Problembeschreibung

Im Rahmen eines universitären Projekts besteht der Zweck dieses Projekts darin, eine große Menge an Textdaten effizient zu verwalten und zu analysieren. Durch die Entwicklung eines Tools sollen Texte effizient analysiert werden, sowie in relevante Kategorien eingeordnet und gemäß festgelegter Kompressionsraten zusammengefasst werden. Das Tool kann beispielsweise von Studierenden genutzt werden, um die Arbeit mit Texten zu vereinfachen.

Projektteilziele und -ergebnisse

Projektteilziele	**Beschreibung**
Auswahl geeigneter Datengrundlagen	Für jede Oberkategorie 700 Datenpunkte
Implementierung einer Pipeline der Datenaufbereitung	Tokenisierung, Stop-Word-Removal, Lemmatisierung, Vektorisierung
Auswählen und Anpassen eines geeigneten Algorithmus	Klassifikation und für Zusammenfassung
Evaluieren der Modelle	Accuracy, F1-Score, F2-Score der Klassifikation. Nachvollziehbare Zusammenfassungen, gemessen auf Einhalten der Kompressionsrate
Nachvollziehbarkeit der Entwicklung mittels GitHub-Commits	Vorhandensein von beschriebenem Code, welche mit passenden Github-Pushs veröffentlicht wurden

Tabelle 1.2: Projektziele und -ergebnisse

Projektgesamtziel

Entwicklung eines Tools zur Textzusammenfassung und -klassifikation. Die Texte sollen unter Einhaltung einer beliebigen Kompressionsrate zusammengefasst werden. Die Klassifi-

kation erfolgt in die Oberkategorien: literarische Texte, wissenschaftliche Abstracts, Rezensionen und Nachrichten.

Nicht-Ziele / Nicht-Inhalte / Projektabgrenzung

- ansprechende, grafische Oberfläche
- ausführlich aufbereitete, deskriptive Analysen
- manuelle Anreicherung der Texte
- Klassifikation von Texten außerhalb der gegebenen Oberkategorien
- Verwendung von vordefinierten Modellen, welche die o.g. Anforderungen bereits komplett abdecken
- Kein Einlesen oder Konvertieren von PDFs

Risiken & Gegenmaßnahmen

Risiken bei Nicht-Durchführung des Projekts wäre eine schlechte Bewertung und im schlimmsten Fall eine Exmatrikulation aus dem Studium. Risiken, die bei der Durchführung des Projekts aufkommen könnten, sind in Tabelle **??** einsehbar, inklusive passender Gegenmaßnahmen.

- Unzureichende Trainingsdaten
  - **Risiko**: Eine unzureichende Menge oder Qualität von Trainingsdaten kann die Leistung des Tools beeinträchtigen
  - **Gegenmaßnahmen**: Es sollte sichergestellt werden, dass genügend qualitativ hochwertige Trainingsdaten gesammelt und verwendet werden. Zudem könnte man auf öffentlich zugängliche Datensätze zurückgreifen, wie Kaggle.
- Personalausfall (Krankheit o.Ä.)
  - **Risiko**: Es besteht das Risiko, dass wichtige Mitglieder des Projekts durch Krankheit oder andere unvorhergesehene Umstände ausfallen könnten. Dies könnte zu Verzögerungen im Projektplan führen und die Qualität der Ergebnisse beeinträchtigen.
  - **Gegenmaßnahme**: Zur Bewältigung dieses Risikos wird ein ausreichender Puffer in der Zeitplanung vorgesehen. Gleichermaßen wird durch die Kompetenz der Teammitglieder versucht, dieses Risiko abzufedern. Darüber hinaus wird durch eine regelmäßige Dokumentation und Kommunikation innerhalb des Teams sichergestellt, dass andere Teammitglieder bei Bedarf Aufgaben übernehmen können.
- Mangelnde Genauigkeit des Modells

  - **Risiko**: Dieses Risiko besteht darin, dass das entwickelte Modell möglicherweise nicht die erforderliche Genauigkeit erreicht, um die Texte effektiv in ihre entsprechenden Oberkategorien zu klassifizieren oder eine zufriedenstellende Zusammenfassung zu liefern. Dies könnte dazu führen, dass das Endprodukt nicht den Erwartungen der Stakeholder entspricht und daher nicht in der geplanten Weise genutzt werden kann
  - **Gegenmaßnahme**: Die Anwendung von Methoden zur Feinabstimmung und Optimierung des Modells ist ein zentraler Ansatz zur Bewältigung dieses Risikos. Dies kann durch die Anpassung und das Training von Hyperparametern erreicht werden, um das Modell zu verbessern. Darüber hinaus ist es wichtig, eine geeignete Evaluierungsmethode zu implementieren, um die Leistung des Modells während des Entwicklungsprozesses kontinuierlich zu überprüfen. So kann festgestellt werden, ob Anpassungen vorgenommen werden müssen, bevor das Modell vollständig implementiert ist. Eine sorgfältige Validierung und Kreuzvalidierung können ebenfalls dazu beitragen, die Genauigkeit des Modells zu verbessern.

- Overfitting
  - **Risiko**: Es besteht die Gefahr, dass das Modell zu sehr auf die Trainingsdaten angepasst wird und nicht gut auf neue, unbekannte Daten reagiert.
  - **Gegenmaßnahme**: Überwachung der Modellperformance sowohl auf Trainings- als auch auf Validierungsdaten und ggf. Anpassung des Modells bzw. der Hyperparameter.
- Zeitdruck
  - **Risiko**: Das Projekt könnte unter Zeitdruck geraten, insbesondere wenn unerwartete Herausforderungen auftreten.
  - **Gegenmaßnahme**: Frühzeitige Projektplanung und Einbeziehung von Puffern für unvorhergesehene Ereignisse.
- Mangel an technischem Know-How
  - **Risiko**: Eine unzureichende Menge oder Qualität von Trainingsdaten kann die Leistung des Tools beeinträchtigen
  - **Gegenmaßnahme**: Bereitstellung passender Dokumentationen und YouTube-Videos für die Teilnehmer, sowie eine Spezialisierung eines Teammitgliedes auf die entsprechenden Inhalte.
- Technische Schwierigkeiten

  - **Risiko**: Während der Entwicklung können unvorhergesehene technische Herausforderungen auftreten, die den Fortschritt behindern.
  - **Gegenmaßnahme**: Einbindung von Experten für technische Unterstützung, Fragen der Betreuer und ausreichende Ressourcen für die Problembehebung und fortlaufende Überwachung des technischen Fortschritts.

Meilensteine

Folgende Meilensteine sind für das Projekt festgelegt:

- 24.05.2023: Abgabe Projektauftrag, Lasten-/Pflichtenheft, grober Strukturplan
- 05.06.2023: Finale Auswahl der geplanten Toolchain, der Algorithmen & geeigneter Daten
- 15.06.2023: Entwicklung einer Pipeline zur Datenaufbereitung
- 22.06.2023: Präsentation: Zwischenstand
- 05.07.2023: Implementieren und Trainieren von den ausgewählten Modellen
- 15.07.2023: Testen und Evaluieren der finalen Modelle
- 27.07.2023: Abschluss des Projekts, inklusive Präsentation der Ergebnisse & Abgabe der Dokumentation

2 Lastenheft

Dieses Lastenheft soll eindeutig definieren, welche Features aus welchem Grund Teil der Software-Lösung „SynTex“ sein müssen und welche nicht. Das Ziel wird erklärt und die Anforderungen an die Software respektive den Algorithmus werden, gruppiert nach verschiedenen Kategorien, genau definiert.

2.1 Ausgangssituation

Im Rahmen eines universitären Projekts besteht die Notwendigkeit, eine große Menge an Textdaten effizient zu verwalten und zu analysieren. Die manuelle Durchführung dieser Aufgaben ist zeitaufwendig und unterliegt menschlichen Fehlern und Einschränkungen. Daher wird ein Textzusammenfassungs- und Klassifikationstool benötigt, um den Prozess der Datenverarbeitung zu automatisieren und zu optimieren. Durch die Implementierung dieses Tools können Texte effizient analysiert, in relevante Kategorien eingeordnet und gemäß festgelegter Kompressionsraten zusammengefasst werden, wodurch der Aufwand für manuelle Datenanalysen erheblich reduziert wird.

2.2 Ziel des Projekts

Das Ziel dieses Projekts ist das Erstellen eines MVPs (Minimum Viable Products) einer Software-Lösung zur Klassifikation & Zusammenfassung von Texten. Eines der beiden größten Ziele des Projekts ist eine möglichst akkurate Textklassifikation folgender Oberkategorien:

- Literarische Texte (inkludiert Kurzgeschichten, Gedichte, Dramen, Essays, (Auto-) Biografien, Märchen und Fabeln)
- Abstracts von wissenschaftlichen Publikationen
- Nachrichten (Zeitungsartikel und Onlineartikel)
- Rezensionen zu Waren und Dienstleistungen

Das zweite Ziel bildet die Erstellung einer Zusammenfassung des gegebenen Texts, wobei die Zusammenfassung um eine beliebige Kompressionsrate X erfolgen soll (also zum Beispiel um 80 %).

2.3 Anforderungen an den Projektablauf & Arbeit mit Stakeholdern

Zeitplanung: Der Projektzeitraum 08.05.2023 - 27.07.2023 ist einzuhalten. Die definierten Meilensteine & Abgaben, zum 24.05.2023, 22.06.2023 und zum 27.07.2023 müssen ebenfalls eingehalten werden:

- 24.05.2023: Abgabe des Projektauftrags und des Lasten-/Pflichtenhefts sowie eines groben Projektstrukturplans, Folien für die Präsentation
- 22.06.2023: Abgabe eines detaillierten Projektstrukturplans und eines Netzplans oder Gantt-Charts, Folien für die Präsentation
- 27.07.2023: Abgabe des Projektabschlussberichts, eine Anführung von Lessonslearned und kritischer Reflexion, Abgabe des finalen Coding inklusive Kommentaren, Folien für die Präsentation

Stakeholder-Beschreibung: Herr Michael Lang und Herr Enzo Hilzinger sind die relevanten Stakeholder in diesem Studienprojekt. Sie haben die Aufgabe der Umsetzung eines Tools für die Textklassifikation und -zusammenfassung an die Studierenden herangetragen, im Rahmen eines Moduls des Studiengangs Wirtschaftsinformatik mit dem Schwerpunkt Data Science. Ihr Interesse liegt darin, den Studierenden Erfahrung im Bereich Projektmanagement sowie in der Projektumsetzung zu geben. Es wird eine umfassende und detaillierte Projektdokumentation und -präsentation erwartet, der Schwerpunkt liegt dabei auf der technischen Umsetzung. Sie dienen als einflussreiche Stakeholder, die jederzeit Änderungswünsche äußern können.

Kommunikation und Stakeholder-Management: Die Kommunikation über wöchentliche Termine mit dem Auftraggeber und den Projektmanagern soll sicherstellen, dass die Erwartungen erfüllt und der Austausch gefördert wird. Die Verfügbarkeit der Projektmanager soll über die angegebenen Kontaktdaten sichergestellt werden.

Change-Management: Durch den regelmäßigen Austausch soll sichergestellt werden, dass Änderungen im Projektumfeld identifiziert, bewertet und gesteuert werden, um sicherzustellen, dass das Projekt auf Kurs bleibt.

Dokumentation, Berichterstattung & Nachvollziehbarkeit: Alle relevanten Informationen und Entscheidungen im Zusammenhang mit dem Projekt müssen dokumentiert werden. Regelmäßige Berichte über den Projektfortschritt und den Status sollten erstellt und an die relevanten Stakeholder kommuniziert werden.

2.4 Schnittstellen:

Eine Schnittstelle für Anwender soll eine simple, grafische Oberfläche sein. Dort sollen die Anwender dann Texte eingeben können sowie die Möglichkeit haben, eine Kompressionsrate für die Zusammenfassung einzustellen. Diese werden dann klassifiziert und zusammengefasst.

2.5 Rand- & Rahmenbedingungen

Es soll ein mehrstufiger Ansatz verfolgt werden. Alle Schritte, von der Datenbeschaffung, -aufbereitung & eventueller -visualisierung bis hin zur geeigneten Auswahl von Modellen, Algorithmen, Bewertungskriterien und Gütekriterien zum Erreichen der genannten Ziele, werden von den Studierenden durchgeführt. Wie in den Anforderungen an den Projektablauf erwähnt, dürfen die genutzten Algorithmen beziehungsweise die verwendeten Modelle nicht auf bereits existierenden vortrainierten oder vordefinierten Tools basieren. Die Wirtschaftlichkeit des Projekts wird komplett ausgeklammert. Es gibt weder ein echtes, noch ein fiktives Unternehmen, aus dessen Sicht in diesem Projekt argumentiert wird. Es handelt sich rein um ein Studienprojekt, welches auch als solches behandelt wird.

2.6 Funktionale Anforderungen an die Software

Die funktionalen Anforderungen an das Tool gestalten sich, wie in den folgenden Sektionen nach dem INVEST-Prinzip definiert. Die wichtigsten Anforderungen betreffen dabei selbstverständlich die Hauptfunktionen, welche die geplante Software erfüllen muss, sowie die gegebene Einschränkung über vordefinierte Tools. Wünschenswert, aber nicht von extremer Bedeutung, sind Anforderungen, welche die Auswahlmöglichkeiten im Tool vereinfachen sowie eine Dokumentation des Tools bereitstellen.

Klassifikation

Das System muss in der Lage sein, den eingegebenen Text präzise in eine der vordefinierten Kategorien einzuordnen. Diese Funktion ist von entscheidender Bedeutung, um die Qualität und Relevanz der durch das System erzeugten Ausgaben zu gewährleisten. Die Funktion sollte auf kleineren, repräsentativen Textbeispielen getestet werden.

Zusammenfassen

Das System muss einen Mechanismus zur Textzusammenfassung unter Verwendung einer variablen Kompressionsrate besitzen. Die Implementierung dieser Funktion ist unerlässlich, um die Benutzerfreundlichkeit des Systems zu erhöhen und die Informationsverarbeitung zu erleichtern.

Unabhängigkeit von vordefinierten Tools

Das System darf keine Abhängigkeiten von vordefinierten Textzusammenfassungs- und Klassifizierungswerkzeugen aufweisen. Dies ist notwendig, um eine höhere Flexibilität bei der

Anpassung und Optimierung des Systems zu gewährleisten und somit ein genaues Verständnis der Funktionsweise zu ermöglichen.

Eingabe des Texts

Es muss ein großes Eingabefeld für Freitexte existieren. Dieses ermöglicht es den Nutzern, beliebige Texte einzugeben, die sie klassifizieren oder zusammenfassen möchten. Das Eingabefeld sollte ausreichend groß sein und dessen Größe, um einen Überblick über die Eingabe zu erhalten. Die Testbarkeit wird durch Prüfung der Eingabe- und Textverarbeitungsfunktionen sichergestellt.

Auswahl der Methode

Das System sollte dem Benutzer die Möglichkeit bieten, auszuwählen, ob er den Text klassifizieren oder zusammenfassen will, oder beides. Diese Funktion verbessert die Flexibilität und Benutzerfreundlichkeit des Systems. Die Umsetzung kann mittels einer grafischen Oberfläche erfolgen.

Dokumentation für Anwender

Es soll eine detaillierte und leicht verständliche Dokumentation des Systems erstellt werden, die sowohl technische Details als auch Benutzeranweisungen enthält. Dies ist von großer Bedeutung, um die Zugänglichkeit und Benutzerfreundlichkeit des Systems zu gewährleisten. Die Verständlichkeit und Vollständigkeit der Dokumentation sollte überprüft werden und die wesentlichen Inhalte der Anwendung beinhalten und ein Verständnis über das Tool hervorbringen.

2.7 Nicht-Funktionale Anforderungen an die Software

Die nicht-funktionalen Anforderungen sind in den folgenden Abschnitten definiert, ebenfalls nach dem INVEST-Prinzip. Die Muss-Anforderungen beziehen sich dabei vor allem auf die Funktionalität der Software, sowie auf die Güte der Modelle & Korrektheit der Ergebnisse. Bei den Soll- und Kann-Anforderungen handelt es sich eher um die Skalierbarkeit & die Fehlerbehebung im Zusammenhang des Tools.

Korrektheit der Ergebnisse

Die Ergebnisse müssen mit den Erwartungen übereinstimmen, um sicherzustellen, dass das System funktioniert wie vorgesehen. Dies stellt einen unabhängigen, verhandelbaren, wertvollen, abschätzbaren, kleinen und testbaren Aspekt des Systems dar. Es ist entscheidend für

die Wertsteigerung des Systems, da korrekte Ergebnisse für den Benutzer von entscheidender Bedeutung sind. Diese Anforderung kann anhand von Testfällen und Metriken verifiziert werden, welche basierend auf die Umsetzung benötigt werden.

Güte der Ergebnisse

Die Güte der Endergebnisse muss mittels verschiedener Kriterien evaluiert werden. Dies ist eine wichtige Anforderung, da sie den Wert des Systems erhöht. Die Güte der Ergebnisse kann durch Vergleich mit Referenzdaten und durch statistische Analyse geschätzt werden. Sie ist testbar und verhandelbar in Bezug auf die spezifischen Gütekriterien.

Funktionalität

Alle im Lastenheft spezifizierten Funktionen müssen ordnungsgemäß implementiert und voll funktionsfähig sein. Dies ist eine zentrale Anforderung, da die Bereitstellung der spezifizierten Funktionen den Nutzen des Systems bestimmt. Die Umsetzung jeder Funktion kann unabhängig geschätzt und getestet werden.

Kompatibilität

Das System muss mit verschiedenen Betriebssystemen (z.B. Windows, macOS, etc.) und Webbrowsern (z.B. Chrome, Firefox, etc.) kompatibel sein. Dies ist eine wichtige Anforderung, um den Nutzwert des Systems für eine breite Palette von Benutzern zu maximieren. Die Kompatibilität kann durch Testen auf verschiedenen Plattformen und Browsern überprüft werden. Sie ist abschätzbar, basierend auf den Anforderungen verschiedener Plattformen und Browser.

Skalierbarkeit

Das System sollte in der Lage sein, horizontal zu skalieren, um den steigenden Benutzeranforderungen gerecht zu werden. Das System sollte in der Lage sein, die Anzahl der gleichzeitigen Benutzer und die Datenverarbeitungskapazität basierend auf den aktuellen Anforderungen dynamisch zu skalieren.

Fehlerbehebung

Das System sollte keine schwerwiegenden Fehler enthalten und bei auftretenden Fehlern angemessen darauf reagieren, um eine korrekte Fehlerbehandlung zu gewährleisten.

Qualitätsmanagement

Das System soll qualitativ hochwertige Ergebnisse liefern, die den definierten Standards und Anforderungen entsprechen.

Erweiterbarkeit

Das System könnte die Möglichkeit bieten, neue Funktionen und Module einfach zu integrieren, um zukünftige Anforderungen und Erweiterungen zu unterstützen.

3 Pflichtenheft

Das vorliegende Pflichtenheft definiert die Anforderungen und Spezifikationen für Modelle zur Textklassifizierung und Zusammenfassung von Texten in den Kategorien wissenschaftliche Texte, literarische Texte, Nachrichten und Rezensionen. Die Modelle werden entwickelt, um eine automatisierte Verarbeitung und Organisation von Textdaten zu ermöglichen und den Benutzern eine effiziente Analyse und Zugänglichkeit zu bieten.

3.1 Auftrag

Die Textklassifikation wird durch den Einsatz eines neuronalen Netzwerks realisiert. Durch die Anwendung eines neuronalen Netzwerks können probabilistische Werte als Ausgabe generiert werden, die eine Einschätzung der Klassifikationssicherheit für den Benutzer ermöglichen. Diese probabilistischen Werte geben an, mit welcher Wahrscheinlichkeit ein Text einer bestimmten Kategorie zugeordnet wird und bieten dem Benutzer eine Einsicht in die Zuverlässigkeit der Klassifikation.
Die generativen Zusammenfassungen basieren auf Transformer-Modellen. Die Länge der Zusammenfassungen kann durch den Benutzer angepasst werden, indem er eine Kompressionsrate wählt. Die Kompressionsrate gibt an, in welchem Maße der Textinhalt komprimiert werden soll, um eine Zusammenfassung zu erstellen. Durch die Wahl einer höheren Kompressionsrate wird der Text stärker komprimiert, während eine niedrigere Kompressionsrate zu einer detaillierteren Zusammenfassung führt.

3.2 Schnittstellen und Kommunikation

Um eine effektive Zusammenarbeit und reibungslose Kommunikation sicherzustellen, werden im Rahmen dieses Projekts klare Schnittstellen und Kommunikationswege definiert. Die Stakeholder, die Dozenten Michael Lang und Enzo Hilzinger, sowie die Projektteilnehmer Niklas Koch, Niclas Cramer, Jasmina Pascanovic und Antoine Fuchs spielen dabei eine entscheidende Rolle. Die Stakeholder sind für die Überwachung des Projekts, die Festlegung der Anforderungen und die Überprüfung der Ergebnisse verantwortlich. Sie kommunizieren Anforderungen und Erwartungen an das System zur Textklassifizierung und Zusammenfassung. Wöchentlich findet ein Austausch zwischen den Stakeholdern und Projektteilnehmern statt. Im Rahmen des Pflichtenhefts werden die folgenden Meilensteine für das Projekt zur Textklassifizierung und Zusammenfassung festgelegt:

1. **24.05.2023:** Abgabe Projektauftrag, Lasten-/Pflichtenheft etc.
   In diesem Meilenstein werden der Projektauftrag sowie das Lasten- und Pflichtenheft

offiziell abgeschlossen und eingereicht. Alle erforderlichen Dokumente und Spezifikationen werden erstellt und überprüft, um eine klare Grundlage für das weitere Vorgehen zu schaffen.

2. **22.06.2023:** Präsentation: Zwischenstand
   Zu diesem Zeitpunkt wird ein Zwischenstand des Projekts präsentiert. Es werden die bisherigen Fortschritte, erreichten Ziele und eventuelle Herausforderungen oder Änderungen vorgestellt. Die Präsentation dient der Überprüfung und Bewertung des bisherigen Projektverlaufs durch die Stakeholder.
3. **27.07.2023:** Abschluss des Projekts inkl. Präsentation & Abgabe der Dokumentation
   Dieser Meilenstein markiert den Abschluss des Projekts. Es erfolgt eine Abschlusspräsentation, in der die endgültigen Ergebnisse, die erreichten Ziele und die erzielten Lösungen präsentiert werden. Gleichzeitig wird die finale Dokumentation, die alle relevanten Informationen, Prozesse und Ergebnisse des Projekts umfasst, abgegeben.

Diese Meilensteine dienen der Projektsteuerung, Überwachung und Dokumentation. Sie stellen wichtige Etappen im Projektverlauf dar und ermöglichen eine gezielte Kontrolle der Fortschritte sowie eine rechtzeitige Kommunikation und Abstimmung mit den Stakeholdern.

3.3 Risikoanalyse: Textklassifizierung und Zusammenfassung

Im Folgenden werden verschiedene Risiken, die bei der Bearbeitung des Projekts auftreten könnten, identifiziert und erklärt. Zudem werden passende Gegenmaßnahmen empfohlen.

Unklare Anforderungen

- **Risiko**: Die Anforderungen an die Textklassifizierung und Zusammenfassung sind unklar oder unvollständig definiert.
- **Auswirkungen**: Missverständnisse bei der Entwicklung des Systems und eine Diskrepanz zwischen den erwarteten und tatsächlichen Funktionen und Leistungen des Systems.
- **Risikobewertung**: Mittel bis hoch.
- **Maßnahmen**: Eine gründliche Anforderungsanalyse und Abstimmung mit den Stakeholdern durchführen, um klare und umfassende Anforderungen zu definieren. Regelmäßige Überprüfungen und Abstimmungen mit den Stakeholdern während des Projekts durchführen.

Technische Herausforderungen

- **Risiko**: Die Implementierung und Integration von neuronalen Netzwerken und Transformer-Modellen kann technisch anspruchsvoll sein. Insbesondere die Hinzunahme einer Kompressionsrate bei der Verwendung von Transformer-Modellen können Herausforderungen entstehen.
- **Auswirkungen**: Schwierigkeiten beim Training, bei der Optimierung der Leistung und Skalierbarkeit des Systems.
- **Risikobewertung**: Mittel bis hoch.
- **Maßnahmen**: Erfahrene und qualifizierte Entwickler mit Kenntnissen im Bereich des maschinellen Lernens und der Datenverarbeitung einbeziehen. Bei Bedarf externe Experten hinzuziehen. Regelmäßige Überprüfung der technischen Umsetzung und gegebenenfalls Anpassung der Vorgehensweise.

Datenqualität und -verfügbarkeit

- **Risiko**: Mangelnde Datenqualität oder begrenzter Zugang zu ausreichenden und repräsentativen Trainingsdaten.
- **Auswirkungen**: Beeinträchtigte Genauigkeit und Zuverlässigkeit der Textklassifizierung und Zusammenfassung.
- **Risikobewertung**: Hoch.
- **Maßnahmen**: Sorgfältige Auswahl und Bereinigung der Trainingsdaten. Gegebenenfalls Ergänzung der vorhandenen Daten durch externe Quellen. Einsatz von Datenverarbeitungstechniken wie Datenaugmentierung, um die Datenmenge zu erweitern.

Performance und Skalierbarkeit

- **Risiko**: Herausforderungen bei der Verarbeitung großer Datenmengen oder Echtzeitverarbeitung.
- **Auswirkungen**: Leistungsprobleme, verzögerte Verarbeitung oder Überlastung des Systems.
- **Risikobewertung**: Mittel bis hoch.
- **Maßnahmen**: Sorgfältige Planung der Systemarchitektur

3.4 Anforderungen: Datenaufbereitung und Beschaffung

Qualitätssicherung der Datengrundlage (Muss): Zur Sicherung der Qualität wird eine gleichverteilte Datenbasis verwendet. Diese Datenbasis beinhaltet rund 700 Instanzen jeder definierten Oberkategorie aus dem Lastenheft. Von diesen 700 Instanzen wird ein Anteil von

etwa 70% für das Training der Modelle verwendet, während die restlichen 30% für das Testen der Modelle eingesetzt werden. Die Gleichverteilung wird verwendet, da somit der Bias minimiert wird, der durch eine unausgewogene Verteilung der Kategorien entstehen könnte. Ist eine Kategorie in der Datenbasis überrepräsentiert, könnte das Modell dazu neigen, diese Kategorie zu bevorzugen und sie möglicherweise fälschlicherweise anderen Kategorien zuzuordnen. Umgekehrt könnte eine unterrepräsentierte Kategorie dazu führen, dass das Modell Schwierigkeiten hat, diese Kategorie korrekt zu erkennen und zuzuordnen. Darüber hinaus wird die inhaltliche Eignung der Datensätze durch vorherige Auswertungen gewährleistet. Dazu werden die vorliegenden Daten einer vorhergehenden Prüfung unterzogen, um ihre Relevanz und Genauigkeit sicherzustellen.

Einlesen der Daten (Muss): Für die Implementierung des Systems wird die Bibliothek *pandas* verwendet, die als Standard für Datenanalyse und -verarbeitung in Python anerkannt ist. Sie dient dazu, verschiedene Arten der Daten einzulesen. Dabei kann dies schlussendlich für genutzten Modelle zur Klassifikation und Textzusammenfassung verwendet werden. Die Verwendung von Pandas ermöglicht ein Einlesen der Datensätze in einem einheitlichen Format, das für die nachfolgenden Verarbeitungsschritte geeignet ist.

Verarbeitung der Daten (Muss): Die initial eingelesenen Daten durchlaufen eine Verarbeitungsphase, in der irrelevante Inhalte wie Satzzeichen oder sogenannte „Stop-Wörter“ mittels der Bibliothek *spaCy* eliminiert werden. Im weiteren Prozess der Datenbearbeitung werden Lemmatisierung und Tokenisierung eingesetzt. Die Lemmatisierung, eine Methode, die Wörter auf ihre Grundform reduziert, und die Tokenisierung, eine Methode, die längere Textstrings in kleinere Einheiten oder Tokenßerlegt, werden zur Nutzung der Daten angewendet. Diese Methoden tragen dazu bei, die Komplexität der Daten zu reduzieren und sie in einem Format bereitzustellen, das für nachfolgende Erstellung der Modelle genutzt werden kann.

3.5 Funktionale Anforderungen der Klassifikation

Unabhängigkeit von vordefinierten Tools (Muss): Zur Sicherstellung der Unabhängigkeit von vordefinierten Tools wird auf ein selbst konfiguriertes neuronales Netzwerk zurückgegriffen, welches für die Durchführung der Textklassifizierungen eingesetzt wird. Dieses angepasste neuronale Netzwerk bietet eine flexible und maßgeschneiderte Lösung für die spezifischen Anforderungen der Textklassifizierungsaufgabe.
Durch die Nutzung eines selbst konfigurierten neuronalen Netzwerkes kann ein hohes Maß an Kontrolle über die Eigenschaften und Parameter des Modells gewährleistet werden. Dies

ermöglicht die präzise Anpassung des Modells an die spezifischen Charakteristiken und Anforderungen des Datensatzes und der Klassifizierungsaufgabe.

Skalierbarkeit (Muss): Die Umsetzung geschieht in einer modularen und flexiblen Architektur. Diese Architektur erlaubt das Hinzufügen, Bearbeiten und Entfernen von Kategorien, ohne die bestehende Funktionalität des Systems zu beeinträchtigen. Die gewählte modulare und flexible Architektur bietet einen entscheidenden Vorteil hinsichtlich der Skalierbarkeit des Systems. Durch diese Designentscheidung wird die Möglichkeit geboten, das System an veränderte Anforderungen anzupassen und zusätzliche Kategorien zu integrieren, ohne dass umfangreiche Änderungen an der bestehenden Struktur erforderlich sind.

Klassifizierung des Inputs (Muss): Im Kontext der Klassifizierung werden Textvektoren eingelesen und gemäß den in dem Lastenheft angegebenen Oberkategorien klassifiziert. Die eingelesenen Textvektoren werden systematisch durch das Klassifizierungsmodell verarbeitet, welches auf den spezifizierten Oberkategorien basiert. Dabei wird eine Wahrscheinlichkeit für die Zugehörigkeit jedes Vektors zu jeder der definierten Kategorien berechnet. Diese probabilistischen Ausgabewerte repräsentieren die Wahrscheinlichkeiten, dass ein bestimmter Textvektor zu den jeweiligen Oberkategorien gehört.

Bestimmung optimaler Hyperparameter (Soll): Im Rahmen der Modelloptimierung werden ausgewählte Hyperparameter durch Hyperparametertuning optimiert. Die zu optimierenden Parameter beinhalten die Lernrate, die Batch-Größe, die Anzahl der Epochen sowie die Anzahl und Größe der verborgenen Schichten.
Im Hyperparametertuning werden verschiedene Kombinationen der genannten Hyperparameter systematisch getestet, um die Konfiguration zu ermitteln, die die beste Leistung des Modells hervorbringt. Dieser Prozess ist entscheidend, um die Genauigkeit und Effizienz des Modells zu maximieren und gleichzeitig das Risiko des Overfittings zu minimieren.
Die Lernrate, die Batch-Größe und die Anzahl der Epochen beeinflussen direkt den Trainingsprozess des Modells, während die Anzahl und Größe der verborgenen Schichten die Komplexität und Kapazität des Modells bestimmen. Durch das Optimieren dieser Hyperparameter kann

3.6 Funktionale Anforderungen der Zusammenfassung

Zusammenfassung des Inputs (Muss): Das Modul nimmt den Text als Input auf und erstellt auf der identifizierten Oberkategorie eine gekürzte Fassung des Originaltextes. Hierfür

werden die relevantesten Kontextinformationen aus den Texten extrahiert, sodass anhand dessen generativ ein neuer Text als Ausgabe erstellt wird.

Unabhängigkeit von vordefinierten Tools (Muss): Für die Umsetzung der Textzusammenfassung wird ein *Transformer-Modell* verwendet. Durch Anwendung von sogenanntem *Transfer-Learning* können die Kontexte der vorliegenden Texte identifiziert werden, um relevante Informationen zu extrahieren. Die Implementierung der verschiedenen Modelle wird in einer Training-Pipeline von *PyTorch* durchgeführt. Auf dieser Grundlage können gehaltvolle Textzusammenfassungen generiert werden. Für jede Oberkategorie wird ein spezialisiertes Modell entwickelt und angewendet.

Evaluation der Sinnhaftigkeit der Zusammenfassung (Muss): Für die Evaluation der inhaltlichen Übereinstimmung der Textzusammenfassungen wird der *ROUGE-Score* genutzt. Dieser Score wird genutzt, da dieser die inhaltliche Kongruenz verschiedener Texte bewertet. Basierend auf diesen Werten wird das beste Modell ausgewählt, welches sich an menschengemachten Zusammenfassungen anpassen.

Bestimmung optimaler Hyperparameter (Muss): Durch Hyperparametertuning werden ausgewählte Hyperparameter (Lernrate, Batch-Größe, Epochenanzahl, Anzahl und Größe der verborgenen Schichten) optimiert. Hierbei werden diese so angepasst, dass optimale Ergebnisse entstehen.

Kompressionsrate (Muss): Anhand der vom Nutzer gewählten Kompressionsrate wird ein Schwellenwert in Form einer Anzahl an Wörtern festgelegt. Auf dieser Basis wird ein Wertebereich festgelegt, welcher durch 5-prozentige Abweichungen von der Kompressionsrate begrenzt ist.

3.7 Allgemeine funktionale Anforderungen

Benutzerfreundlichkeit (Muss): Die Benutzerfreundlichkeit des Systems wird durch die Implementierung einer Funktion gewährleistet, die die ausgewählten Dokumente, die Kompressionsrate und die Kategorien als Eingabewerte akzeptiert. Diese Funktion fasst alle Schritte - Preprocessing, Klassifikation und Zusammenfassung - des mehrstufigen Ansatzes zusammen und führt sie in einer gebündelten Operation aus. Die am besten trainierten Modelle der Klassifikation und Zusammenfassung werden jeweils für die Pipeline verwendet.

Input des Users (Soll): Die Benutzerinteraktion erfolgt über eine Web-Darstellung, in der Eingaben über diverse Eingabefelder ermöglicht werden. Diese Felder umfassen Check-Boxen für die Wahl der benötigten Leistung wie Klassifikation und Zusammenfassung, einen Slider für die Einstellung der Kompressionsrate und ein Freifeld für die Texteingabe.

Erweitere Darstellung der Ergebnisse (Kann): Die Ergebnisse des Systems werden in Form einer Web-Darstellung präsentiert. Hierbei werden die Outputs sowohl in Form einer visuellen Wahrscheinlichkeitsverteilung der Klassen als auch als zusammengefasster Text dargestellt. Die Wahrscheinlichkeitsverteilung der Klassen wird visuell dargestellt, um den Nutzern eine intuitive und leicht verständliche Darstellung der Klassifikationsergebnisse zu bieten. Zusätzlich zur Darstellung der Klassifikationsergebnisse wird auch der zusammengefasste Text ausgegeben. Dies ermöglicht den Nutzern, den Text in seiner komprimierten Form zu betrachten und dessen Inhalt und Bedeutung im Kontext der Klassifikationsergebnisse zu interpretieren.

Nachvollziehbarkeit der technischen Bestandteile (Muss): Die Nachvollziehbarkeit der technischen Komponenten wird durch das Hinzufügen von Kommentaren zum vorliegenden Code sichergestellt. Im gesamten Code sind erläuternde Kommentare eingebettet, die den Zweck und die Funktionalität der jeweiligen Codeabschnitte verdeutlichen. Diese Kommentare bieten den Nutzern einen Einblick in die technische Umsetzung der verschiedenen Komponenten des Systems.

Nachvollziehbarkeit des Entwicklungsprozesses (Muss): Die Nachvollziehbarkeit des Entwicklungsprozesses wird durch die Dokumentation des Projekts auf der Plattform *Github* sichergestellt. Hierbei werden sämtliche Codeänderungen und -erweiterungen von den Projektteilnehmern durch Commits dokumentiert, wodurch eine nachvollziehbare Timeline des Entwicklungsprozesses erzeugt wird.

Skalierbarkeit (Muss): Die Skalierbarkeit der Anwendung wird durch die Strukturierung der Module als separate Funktionen gewährleistet, welche eine wiederverwendbare Nutzung ermöglichen. Hierbei können neue Funktionen hinzugefügt, bestehende Funktionen modifiziert oder entfernt werden, ohne die Gesamtfunktionalität des Systems zu beeinträchtigen.

Fehlerprävention (Soll): Im Rahmen des Systems werden Fehlermeldungen genutzt, um aufgetretene Probleme oder Abweichungen vom erwarteten Verhalten zu identifizieren und zu klassifizieren. Diese Meldungen liefern detaillierte Informationen über die Art des aufgetretenen Fehlers und den Kontext, in dem er aufgetreten ist. Auf Grundlage dieser Informa-

tionen können gezielte Anpassungen vorgenommen werden, um den Fehler zu beheben und ähnliche Probleme in der Zukunft zu vermeiden.

3.8 Nicht-Funktionale Anforderungen

Leistung (Muss): Bei der Konzeption des Modells liegt der Fokus auf dessen Kompaktheit, um eine schnelle Verarbeitung und damit geringe Laufzeiten zu erreichen. Durch den bewussten Verzicht auf überflüssige oder ineffiziente Komponenten wird das Modell so gestaltet, dass es seine Aufgaben mit maximaler Effizienz ausführen kann.

Übersichtlichkeit des Codes (Soll): Im Designprozess des Systems wird besonderer Wert auf eine modulare Gestaltung der Codebestandteile gelegt. Durch die Unterteilung des Codes in eigenständige Module wird eine klare und strukturierte Darstellung des Codes erreicht, die die Übersichtlichkeit und Verständlichkeit des Codes verbessert. Diese Funktionen und die dazugehörigen Variablen werden so gestaltet, dass eine intuitive Nutzung des Anwenders ermöglicht wird.

Kompatibilität (Soll): Die Kompatibilität der Anwendung wird durch Durchführung von Tests auf verschiedenen Betriebssystemen und anschließender Anpassung des Codes gewährleistet, um auf jedem der getesteten Systeme volle Funktionalität zu gewährleisten.