Duale Hochschule Baden-Württemberg Mannheim

Dokumentation

Entwicklung eines NLP-Tools zur Textzusammenfassung und -klassifizierung

Studiengang Wirtschaftsinformatik

Studienrichtung Data Science

Dozenten:	Michael Lange, Enzo Hilzinger
Verfasser:	Bonn Lukas, Amina Uicker-Darwish, Jan Rüdt, Aymane Bougern
Matrikelnummer:	5856761, 8278962, 1737304, 1552312
Kurs:	WWI20DSA
Studiengangsleiter:	Prof. Dr.-Ing. habil. Dennis Pfisterer
Bearbeitungszeitraum:	08.05.2023 – 24.05.2023

Inhaltsverzeichnis

1 Projektauftrag

Projektname: Entwicklung eines NLP-Tools für die Klassifikation und das Zusammenfassen von Texten.

Auftraggeber: Das Projekt findet im Rahmen eines Hochschulprojekts statt. Der Projektauftrag geht von den Dozierenden der Vorlesung „Projektrealsierung“ aus.

Projektausrichtung: Basierend auf den Informationen und den Anforderungen des Auftraggebers, ist das Projekt der Kategorie Textverarbeitung zuzuordnen wofür auch das Projektmanagement eine wichtige Rolle spielt. Das Projekt ist in erster Linie ein Forschungs- und Bildungsprojekt und zielt nicht direkt auf finanzielle Ergebnisse wie eine Umsatzsteigerung ab.

Ausgangssituation: Die Ausgangssituation für das Projekt ist die Notwendigkeit eines leistungsfähigen NLP-Tools zur Textklassifikation und Zusammenfassung. Das manuelle Klassifizieren und Zusammenfassen von Texten kann zeitaufwendig, fehleranfällig und ineffizient sein. Es besteht ein Bedarf an einer automatisierten Lösung, die eine schnelle und präzise Klassifikation und Zusammenfassung von Texten ermöglicht.

Projektziel: Das Projektziel besteht darin, ein NLP-Tool zur Textklassifikation und -zusammenfassung zu entwickeln. Unter Vorbehalt vorhandener Datensätze soll das Tool in der Lage sein, verschiedene Nachrichtenartikel automatisch zu klassifizieren und anhand einer frei wählbaren Kompressionsrate zusammenzufassen. Werden keine passenden Datensätze gefunden, kann dies angepasst werden.

Beteiligte Personen: Da es sich um ein Forschungsprojekt der Hochschule handelt, welches im Rahmen eines Moduls erarbeitet werden soll, sind die Studenten hauptsächlich für die Umsetzung verantwortlich. Daher sind keine anderen Abteilungen oder Bereiche in dem klassischen Sinne mit involviert.

Projektinhalt:

- **Entwicklung eines NLP-Tools:** Das Projekt zielt darauf ab, ein leistungsfähiges und funktionsfähiges NLP-Tool zur automatisierten Textklassifikation und Zusammenfassung zu entwickeln.

- **Textklassifikation:** Das Tool soll in der Lage sein, Texte in verschiedene vordefinierte Kategorien einzuteilen. Durch die Analyse des Textinhalts und anderer Merkmale soll das Tool in der Lage sein, die richtige Kategorie für jeden Text zu bestimmen.
- **Textzusammenfassung:** Das Tool soll den Originaltext unter Berücksichtigung einer gewünschten Kompressionsrate zusammenfassen. Dabei sollen wichtige Informationen und Kernaspekte beibehalten werden, während unwichtige Informationen entfernt werden.

Projektabgrenzung:

- **Sprachliche Begrenzung:** Das Tool wird sich auf die Verarbeitung von Texten in der englischen Sprache konzentrieren.
- **Textarten:** Das Tool wird in der Lage sein, verschiedene Arten von Texten zu verarbeiten, wie z.B. Artikel und andere Texte. Es kann jedoch auf bestimmte Textsorten beschränkt sein, abhängig von den verfügbaren Trainingsdaten und der Implementierung.
- **Kategorien:** Das Tool kann vordefinierte Kategorien für die Textklassifikation unterstützen. Die Anzahl und Art der Kategorien kann jedoch begrenzt sein und muss im Rahmen des Projekts definiert werden.

Mögliche Risiken und Gegenmaßnahmen:

- Unzureichende Textdaten für die Klassifikation und Zusammenfassung
  - Eine gründliche Planung und Beschaffung von qualitativ ausreichenden Textdaten
- Komplexität der NLP-Algorithmen und Herausforderungen bei der Modelloptimierung
  - Eine umfassende Literaturrecherche und Analyse bewährter Praktiken in der NLP-Forschung sowie ein enger Austausch mit Experten zu diesen Themen
- Keine rechtzeitige Lieferung des Tools mit den dazugehörigen Komponenten
  - Eine detaillierte Projektplanung mit realistischen Zeitrahmen für jede Phase des Projekts
- Unzureichende Dokumentation des entwickelten Tools und der Algorithmen

  - Kontinuierliche Dokumentation im Rahmen des gesamten Projektverlaufs, einschließlich den Designentscheidungen, den Implementierungsdetails und den erzielten Ergebnissen. Erstellung einer umfassenden Dokumentation, um die Wartbarkeit und Weiterentwicklung des Tools zu erleichtern.

Risiken bei Nichtdurchführung des Projekts:

- **Verbleibende Ineffizienz manueller Prozesse:** Ohne das NLP-Tool müssten Texte weiterhin manuell klassifiziert und zusammengefasst werden, was zeitaufwendig, fehleranfällig und ineffizient sein kann.
- **Begrenzte Skalierbarkeit:** Ohne das Tool könnte es schwierig sein, mit der steigenden Anzahl von Texten und der Nachfrage nach Klassifikations- und Zusammenfassungstools Schritt zu halten.

Nutzen des Projekts:

- **Zeitersparnis:** Das Tool ermöglicht eine automatisierte Textklassifikation und -zusammenfassung, was zu erheblichen Zeitersparnissen führen kann.
- **Steigerung der Produktivität:** Das Tool ermöglicht eine schnellere und effizientere Verarbeitung von Texten, was zu einer Steigerung der Produktivität führen kann.
- **Kosteneinsparungen:** Durch die Automatisierung des Prozesses können Kosten für manuelle Arbeitsstunden eingespart werden.

Voraussetzung / Rahmenbedingungen / Abhängigkeiten:

- **Verfügbarkeit von Textdaten:** Um das NLP-Tool zu entwickeln und zu trainieren, werden umfangreiche Textdaten in den gewünschten Kategorien benötigt.
- **Zugriff auf NLP-Methoden und -Algorithmen:** Das Projekt erfordert Zugang zu modernsten NLP-Methoden und -Algorithmen sowie zu entsprechenden Bibliotheken oder Frameworks. Eine gute Kenntnis der NLP-Prinzipien und der einschlägigen Technologien ist erforderlich.
- **Expertise in maschinellem Lernen:** Da das NLP-Tool voraussichtlich zumindest teilweise auf maschinellem Lernen basiert, ist fundiertes Wissen und Erfahrung in diesem Bereich erforderlich.

2 Lasten- und Pflichtenheft

2.1 Lastenheft

2.1.1 Einleitung und Hintergrundinformationen

Das vorliegende Lastenheft beschreibt die Anforderungen und Erwartungen an das zu entwickelnde NLP-Tool. Es dient als Grundlage für die Spezifikation des Projekts und legt die Rahmenbedingungen für die Umsetzung fest. Das Lastenheft dient als Instrument zur Zusammenarbeit zwischen dem Auftraggeber und dem Auftragnehmer (vgl. Korte et al., 2021, S. 7). In diesem werden die Anforderungen des Auftraggebers an die Lieferung und die zu erbringende Leistung definiert, wobei sowohl die funktionalen als auch die nichtfunktionalen Anforderungen von Relevanz sind (vgl. Korte et al., 2021, S. 7; Asif et al., 2019, S. 36164).

Heutzutage steht eine große Menge an Texten zur Verfügung, die es zu verarbeiten und zu verstehen gilt. Allerdings kann es zeitaufwendig sein, relevante Informationen aus Texten zu extrahieren oder Texte verschiedenen Kategorien zuzuordnen.

Aus diesem Grund wurde seitens des Auftraggebers beschlossen ein NLP-Tool zu entwickeln, das diese Herausforderungen angeht und eine effizientere Textverarbeitung ermöglicht. Das Tool soll Benutzern helfen, Zeit zu sparen und den Überblick über große Textmengen zu behalten.

Der Auftrag besteht darin, ein NLP-Tool zu entwickeln, welches eine Textzusammenfassung und -klassifikation ermöglicht. Das Tool soll in der Lage sein, Texte basierend auf einer benutzerdefinierten Kompressionsrate zusammenzufassen und sie automatisch verschiedenen vordefinierten Kategorien zuzuordnen. Das Ziel ist es, durch eine effiziente Textzusammenfassung den Überblick über große Textmengen zu ermöglichen und aufbauend auf der präzisen Textklassifikation die Suche nach bestimmten Textsorten zu erleichtern. Insgesamt erleichtert das Tool die Informationsverwaltung von Texten und gleichzeitig spart der Benutzer Zeit.

2.1.2 Funktionale Anforderungen

Die funktionalen Anforderungen an das zu entwickelnde Tool zur Textzusammenfassung und -klassifikation umfassen eine Vielzahl von Funktionen, die es Benutzern ermöglichen, Texte effizient zusammenzufassen, zu klassifizieren und wichtige Informationen extrahiert zu erhalten. Das Tool soll folgende Funktionalitäten bieten:

- **Zusammenfassungsfunktion:** Eine wesentliche Anforderung an das Tool besteht darin, Texte in englischer Sprache automatisch zusammenzufassen. Dabei sollte den Benutzern die Möglichkeit geboten werden, eine gewünschte Kompressionsrate anzugeben, beispielsweise 80%. Die Zusammenfassungsfunktion des Tools hat die Aufgabe, die wesentlichen Informationen aus dem Text zu extrahieren und sie in einer prägnanten Form darzustellen. Es ist von großer Bedeutung, dass die Zusammenfassung verständlich und grammatikalisch korrekt ist, um den Benutzern eine klare und gut strukturierte Zusammenfassung des Textinhalts zu liefern.
- **Klassifikationsfunktion:** Eine weitere wichtige Anforderung an das Tool besteht darin, gegebene Texte in englischer Sprache in verschiedene Oberkategorien zu klassifizieren. Diese Oberkategorien sollen sich auf Nachrichtenartikel fokussieren, wobei dies bei unzureichender Datengrundlage angepasst werden kann. Die Klassifikationsfunktion des Tools soll präzise und zuverlässig sein, um eine genaue Zuordnung der Texte zu ermöglichen.
- **Bereitstellung einer Benutzerschnittstelle:** Das NLP-Tool soll eine Benutzerschnittstelle beinhalten, die es ermöglicht, die Klassifikations- und Zusammenfassungsfunktion dem Benutzer zur Verfügung zu stellen. Die Schnittstelle sollte in der Lage sein, Texte entgegenzunehmen und die entsprechenden Zusammenfassungs- und Klassifikationsfunktionen auf diesen anzuwenden. Dabei soll der Benutzer die Möglichkeit haben, eine gewünschte Kompressionsrate anzugeben, welche für die Länge des Ausgabetexts berücksichtigt wird. Zudem sollten die Resultate dem Benutzer angezeigt werden können.
- **Mehrstufiger Ansatz:** Das Tool soll einen mehrstufigen Ansatz zur Realisierung der Zusammenfassungs- und der Klassifikationsfunktion verfolgen. Dieser Ansatz kann beispielsweise Schritte innerhalb der Data Selection, der Vorverarbeitung und des Modelltrainings beinhalten. Es ist von großer Bedeutung, eine geeignete Pipeline, einen passenden Algorithmus oder vortrainierte Modelle auszuwählen und anzupassen, um verhältnismäßig gute Ergebnisse bei der Zusammenfassung und Klassifikation von

Texten zu erzielen. Es sollte zudem eine flexible Architektur gewährleistet werden, um zukünftige Modelle oder Funktionalitäten in das Tool zu integrieren.

- **Eigenständige Auswahl der Toolchain:** Die Auswahl der Tools sollte basierend auf den spezifischen Anforderungen des Projekts, den Fähigkeiten des Entwicklungsteams und den verfügbaren Ressourcen getroffen werden. Dabei sollten Tools ausgewählt werden, welche die gestellten Anforderungen unterstützen und gleichzeitig eine effektive und skalierbare Entwicklung ermöglichen. Natürliche Sprachverarbeitungstools, wie NLTK (Natural Language Toolkit), spaCy oder ähnliche, sowie Machine Learning-Frameworks, wie TensorFlow, PyTorch oder scikit-learn, gehören zu den erlaubten Tooltkits. Allerdings sollen keine fertigen Tools oder Lösungen verwendet werden, die bereits alle erforderlichen Funktionen für die Textzusammenfassung und -klassifikation bereitstellen.

2.1.3 Nicht-funktionale Anforderungen

Neben den funktionalen Anforderungen gibt es auch nicht-funktionale Anforderungen an das zu entwickelnde NLP-Tool. Diese Anforderungen beschreiben die Eigenschaften oder Rahmenbedingungen des Tools, welche über die reinen Funktionalitäten hinausgehen (vgl. Kusay-Merkle et al., 2018, S. 362). Folgende nicht-funktionale Anforderungen sollen durch das Tool erfüllt werden:

- **Leistung:** Die Zusammenfassungs- und Klassifikationsfunktionen sollten in angemessener Zeit ausgeführt werden, um einen reibungslosen Betrieb zu gewährleisten.
- **Benutzerfreundlichkeit:** Das Tool sollte für die Benutzer einfach und unkompliziert in der Handhabung sein. Die Benutzeroberfläche sollte so gestaltet sein, dass sich die Benutzer schnell zurechtfinden können, ohne umfangreiche Schulungen oder Anleitungen zu benötigen.
- **Nachvollziehbarkeit:** Der gesamte Entwicklungsprozess sollte nachvollzogen werden können, sodass der Auftraggeber die im Rahmen der Entstehung des NLP-Tools durchgeführten Schritte verstehen kann.

2.2 Pflichtenheft

Das Ziel des Textzusammenfassungs- und Klassifikationstools ist es, dem Benutzer eine effiziente Verarbeitung von Texten zu ermöglichen. Durch die automatische Zusammenfassung von Texten anhand einer Komperessionsrate und die Zuordnung zu vordefinierten Kategorien soll der Benutzer in der Lage sein, den Überblick über relevante Informationen zu behalten und den Text gezielt in bestimmte Textsorten klassifizieren zu können. Dabei werden fortschrittliche NLP-Techniken eingesetzt, um die im Lastenheft definierten Anforderungen zu erfüllen.

2.2.1 Projektmanagement

Projektorganisation

Im Rahmen der Entwicklung des NLP-Tools und der damit verbundenen Aufgaben wird ein Team von vier Mitgliedern gestellt. Alle beteiligten Teammitglieder sind aktuelle Studenten im Bereich Wirtschaftsinformatik - Data Science und verfügen über fundierte Kenntnisse und Erfahrungen in diesem Fachgebiet. Durch die vorherige Teilnahme an NLP-Projekten während des Studiums haben die Personen ein solides Verständnis für die technischen und organisatorischen Aspekte solcher Projekte entwickelt. Auch verfügen die Projektteilnehmer über Kenntnisse im Umgang mit dem Web-Framework Flask und mit der Programmiersprache Python. Jedes Teammitglied wird mit gleichen Anteilen an der Umsetzung des Projekts beteiligt sein und bringt ähnliche Kompetenzen und Fähigkeiten in das Projekt ein, um eine effektive Umsetzung zu gewährleisten. Ein Projektmitglied wurde als Projektleiter ernannt und ist damit sowohl für die Kommunikation mit den Auftraggebern als auch für die aktive Mitentwicklung des NLP-Tools verantwortlich. Neben dem Projektleiter konzentrieren sich die drei weiteren Teammitglieder intensiv auf die Entwicklung des Tools. Eine genauere Aufgabeneinteilung erfolgt im späteren Projektverlauf mit der Beschreibung der Arbeitspakete im Rahmen der Erstellung des detaillierten Projekttrukturplans. Durch die enge Zusammenarbeit und den Wissensaustausch innerhalb des Teams kann jedes Teammitglied seine individuellen Stärken und Kompetenzen optimal einbringen.

Projektmeilensteine

Das Projekt umfasst einen Zeitraum von 11,5 Wochen und muss vor dem 27.07.2023 abgeschlossen sein. Es wurde eine strukturierte Vorgehensweise mit mehreren Meilensteinen festgelegt, um eine systematische Durchführung des Projekts zu gewährleisten. Jeder Meilenstein hat einen Endtermin, welcher durch den Auftraggeber festgelegt wurde. Das Projekt beginnt mit der groben Planung und Formulierung des Projektauftrags, gefolgt von der detaillierten Planung und Definition der Arbeitspakete sowie der eigentlichen Toolentwicklung. Abschließend werden der Projektabschlussbericht erstellt und wichtige Erkenntnisse aus dem Projekt dokumentiert. Durch diese Einteilung wird ein geordneter und effizienter Projektablauf gewährleistet. Folgende Meilensteine sind dabei im Rahmen des Projekts definiert:

Abgabefristen / Meilensteine	**Laufzeit / Endtermin**	**Beschreibung**
Meilenstein 1: Erste Planung	24.05.2023	Formulieren des Projektauftrags, Erstellen des Lasten- und Pflichtenhefts und Entwerfen des groben Projektstrukturplans
Meilenstein 2: Detaillierte Planung	22.06.2023	Entwerfen des detaillierten Projektstrukturplans und Gantt-Charts
Meilenstein 3: Toolentwicklung und Abschlussprojektberichterstellung	27.07.2023	Entwickeln der Textzusammenfassung, -klassifikation und Webanwendung, Erstellen des Projektabschlussberichts, Erarbeiten der Lessons Learned und Durchführen einer kritischen Reflexion

Tabelle 2.1: Abgabefristen und Meilensteine des Projekts

Im Verlauf des Projekts wird eine detaillierte Arbeitsaufteilung im Rahmen des detaillierten Projektstrukturplans vorgenommen, um eine präzise Steuerung und Umsetzung zu gewährleisten. Hierbei werden auch die Arbeitspakete definiert. Zudem erfolgt die Erstellung eines

Gantt-Diagramms, um einen übersichtlichen Zeitplan mit konkreten Terminen und Abhängigkeiten darzustellen. Diese Strukturierung ermöglicht eine effiziente Ressourcenplanung, eine klare Zuweisung von Aufgaben und eine präzise Zeitplanung für jedes Arbeitspaket.

Projektmanagementmethode und -vorgehen

Im Rahmen dieses Projekts empfiehlt sich die Verwendung einer klassischen Projektmanagementmethode, insbesondere des Wasserfallmodells. Das Wasserfallmodell zeichnet sich durch seine sequenzielle Vorgehensweise aus, bei der die einzelnen Meilensteine des Projekts nacheinander abgearbeitet werden. Angesichts der festen Fristen für jeden Meilenstein ist diese Methode gut geeignet, um den Projektfortschritt und die Einhaltung der Zeitpläne zu überwachen (vgl. Vieweg, 2015, S. 41f).

Da das Projekt klare und vordefinierte Ziele, Anforderungen und Meilensteine hat, bietet das Wasserfallmodell eine strukturierte Vorgehensweise, um diese Schritte systematisch umzusetzen. Es ermöglicht eine genaue Planung, Ressourcenallokation und Zeitrahmenfestlegung für jede Phase des Projekts. Die klassische Wasserfallmethode bietet auch klare Strukturen für die Kommunikation mit den Stakeholdern. Der Projektleiter kann den Stakeholdern regelmäßig Fortschrittsberichte liefern und Feedback einholen, um sicherzustellen, dass das Projekt den Anforderungen entspricht. Es ist jedoch zu beachten, dass die Wasserfallmethode weniger flexibel ist, wenn es um Anpassungen oder Änderungen während des Projekts geht. Wenn sich während der Durchführung des Projekts Anforderungen ändern oder neue Informationen auftauchen, kann es schwierig sein, diese in die bereits abgeschlossenen Phasen zu integrieren (vgl. Vieweg, 2015, S. 41f).

Die Zeitplanung ist ein wichtiger Bestandteil des Projektmanagements. Der Zeitrahmen für die Toolerstellung und Dokumentation wird im detaillierten Projektstrukturplan festgelegt. Der Zeitplan sollte realistisch sein und ausreichend Puffer für unvorhergesehene Ereignisse und Änderungen berücksichtigen. Die Dokumentation spielt ebenfalls eine bedeutende Rolle im Projektmanagement. Diese Dokumentation dient als Referenz für alle Projektbeteiligten und ermöglicht eine klare Kommunikation und ein Verständnis der Projektanforderungen. Die fortschreitende Dokumentation erfolgt dabei über die Plattform Github, was in Unterabschnitt 2.2.3 beschrieben wird. Das Endergebnis wird rückblickend in einem Projektabschlussbericht dokumentiert und bewertet. Darin wird die Projektumsetzung analysiert, Stärken und Verbesserungspotentiale aufgezeigt sowie Lessons Learned und Empfehlungen für zukünftige Projekte dokumentiert. Abschließend findet eine Projektab-

nahme seitens des Auftraggebers statt, bei der die Erfüllung der definierten Anforderungen und die Zielerreichung gemäß dem Pflichtenheft überprüft werden.

2.2.2 Systemarchitektur

Die Systemarchitektur des Textzusammenfassungs- und Klassifikationsstools basiert auf einer modularen Struktur, die verschiedene Komponenten umfasst.

Die Architektur umfasst zunächst die Bereitstellungsumgebung in Form einer Webanwendung, welche als Schnittstelle zwischen dem Benutzer und den Backend-Komponenten dient, um eine nahtlose Interaktion und Verarbeitung der Texte zu gewährleisten. Sie ermöglicht es dem Benutzer, Texte einzugeben, die gewünschte Kompressionsrate anzugeben und die Ergebnisse der Zusammenfassung und Klassifikation anzuzeigen. Die Benutzerschnittstelle wird dazu mit den benötigten NLP-Funktionen verbunden.

Durch die Textzusammenfassungskomponente kann der eingegebene Text über eine Funktion analysiert und zum Erstellen einer prägnanten Zusammenfassung genutzt werden. Dies ermöglicht dem Benutzer einen schnellen Überblick über den Inhalt des Textes und eine präzise Extraktion relevanter Informationen (vgl. Gupta, Lehal et al., 2009, S. 62).

Parallel dazu erfolgt die Textklassifikation durch eine separate Komponente, welche auf NLP-Techniken basiert. Hierbei können verschiedene Modelle und Funktionen zum Einsatz kommen, um den Text basierend auf seinem Inhalt verschiedenen vordefinierten Klassen zuzuordnen (vgl. Gupta, Lehal et al., 2009, S. 63).

Zusätzlich bietet die Architektur die Möglichkeit zur Integration weiterer Modelle und Funktionalitäten.

Die Modelle und Funktionen bilden die Kernkomponenten der Architektur. Sie werden auf Basis von ausgewählten Datensätzen trainiert und evaluiert, um eine präzise Extraktion relevanter Informationen sowie eine qualitativ hochwertige Textklassifikation zu ermöglichen.

2.2.3 Toolentwicklung im Überblick

Die Toolentwicklung basiert auf der zuvor dargestellten Systemarchitektur und gliedert sich in drei Hauptteile:

1. **Textzusammenfassung**
2. **Textklassifikation**
3. **Webanwendung**

Im folgenden Abschnitt werden diese Teile einzeln in separaten Unterabschnitten detailliert in Bezug auf die technische Umsetzung beschrieben. Dies ermöglicht eine umfassende Darstellung der einzelnen Komponenten und ihrer Implementierung im Gesamtkontext des Tools.

Wie bereits beschrieben, wird im Rahmen des Entwicklungsprozesses GitHub als zentrale Plattform genutzt, um eine umfassende Dokumentation aller Entwicklungsschritte bereitzustellen. Dadurch wird sichergestellt, dass sämtliche Änderungen und Fortschritte transparent und nachvollziehbar erfasst werden und allen Beteiligten, einschließlich dem Auftraggeber, dargelegt werden können, um so die Anforderung an die Nachvollziehbarkeit der Toolentwicklung zu erfüllen.

2.2.4 Textzusammenfassung

Implementierung

Die Umsetzung der Textzusammenfassung erfolgt durch die Anwendung eines *„extractive"*-Ansatzes. Dies ist eine Methode, bei der die relevanten Sätze aus dem Text extrahiert und zu einer Zusammenfassung kombiniert werden. Im Gegensatz dazu werden bei einem sogenannten *„abstractive"*-Ansatz die grundlegenden Informationen aus dem Originaltext extrahiert, wobei die Sätze innerhalb der Zusammenfassung nicht Teil des Originaltexts sein müssen (vgl. Mithbavkar & Chauhan, 2021).

Innerhalb des gegebenen Projekts wurde sich für einen *„extractive"*-Ansatz entschieden, um dadurch insbesondere vor dem Hintergrund einer Kompressionsrate im Rahmen des Projektzeitraums die Flexibilität zu haben, die Satzanzahl ohne großen Aufwand unter Berücksichtigung der Kompressionsrate anzupassen. Zudem kann durch diesen Ansatz die grammatikalische Korrektheit des Textes gewährleistet werden, solange auch die Sätze des Originaltexts über eine korrekte Grammatik verfügen.

Um die Zusammenfassung zu ermöglichen, durchläuft der Text verschiedene Vorverarbeitungsschritte. Der Text wird dabei in die einzelnen Sätze tokenisiert. Auch können im

Rahmen der Vorverarbeitung Stoppwörter in dem Text entfernt werden (vgl. Mazumdar, 2023).

Im Zuge der Textzusammenfassung sind insbesondere drei Phasen von Relevanz. In der ersten Phase können der Eingabetext und die einzelnen Sätze darin auf verschiedene Weisen repräsentiert werden, um dadurch die Relevanz der Sätze zu ermitteln. Beispielsweise können dabei Wörter extrahiert werden, welche das Thema des Texts bestmöglich beschreiben, sodass Sätze in denen diese Wörter auftreten als wichtiger erachtet werden als andere *(Topic words)*. Auch kann die Länge der Sätze oder die Position dieser im Dokument einen Einfluss auf die Relevanz der Sätze haben. In der darauffolgenden Phase werden die einzelnen Sätze auf Basis der Erkenntnisse aus der ersten Phase gewichtet, sodass als relevant erachtete Sätze eine höhere Gewichtung erhalten als weniger relevante. In Phase drei werden anschließend die Sätze mit der höchsten Gewichtung dazu genutzt, die Zusammenfassung zu erstellen (vgl. Sciforce, 2019).

Innerhalb des betrachteten Projekts wird im Zuge der dritten Phase zusätzlich die Kompressionsrate genutzt, um die Anzahl der Sätze zu ermitteln, welche im Rahmen der Zusammenfassungserstellung verwendet werden sollen. Dadurch soll die diesbezügliche Anforderung erfüllt werden. Je nach vom Benutzer eingegebenen Kompressionsrate werden mehr oder weniger Sätze der zu erstellenden Zusammenfassung hinzugefügt.

Datenbeschaffung

Im Rahmen der Realisierung wird eine Zusammenfassungsfunktion unter Berücksichtigung des *„extractive“*-Ansatzes entwickelt. Um die Anforderung an die Textzusammenfassung zu erfüllen wird zudem eine diesbezügliche Evaluierung durchgeführt, weshalb hierfür geeignete Daten benötigt werden.

Ein passender Datensatz enthält verschiedene Texte mit einer gewissen Länge, sodass diese sich für eine Zusammenfassung eignen. Zusätzlich muss der Datensatz Zusammenfassungen enthalten, die mit den entsprechenden Texten verknüpft sind. Für die Aussagekraft der Evaluierung ist es von Vorteil, wenn die bereitgestellten Zusammenfassungen im Datensatz ebenfalls auf Basis eines *„extractive“*-Ansatzes erstellt wurden. Dies gewährleistet eine konsistente Methodik und ermöglicht eine erfolgreiche Evaluierung des Tools.

Darüber hinaus müssen die Texte in englischer Sprache verfasst sein, um dadurch die Zusammenfassung dieser zu ermöglichen.

Die Verfügbarkeit wie auch die Zugänglichkeit des Datensatzes sind ebenfalls wichtige Aspekte. Ein idealer Datensatz ist frei und öffentlich zugänglich, um sicherzustellen, dass er für den gegebenen Anwendungsfall verwendet werden darf.

Evaluierung

Die Evaluierung der Zusammenfassungsfunktion erfolgt auf der Grundlage von spezifischen Metriken, welche darauf abzielen, die Qualität der generierten Zusammenfassungen zu bewerten.

Eine beispielhafte Metrik ist ROUGE *(Recall-Oriented Understudy for Gisting Evaluation)*. ROUGE bewertet die Übereinstimmung einer automatisch generierten Zusammenfassung mit einer Referenzzusammenfassung anhand von Übereinstimmungsmetriken wie ROUGE-N (basierend auf der N-Gramm-Übereinstimmung), ROUGE-L (basierend auf der längsten gemeinsamen Teilsequenz) und ROUGE-S (ermöglicht das Ignorieren einzelner eingeschobener Worte im Zuge des Prüfens der Übereinstimmung) (vgl. Chiusano, 2022b).

Eine weitere Metrik ist BLEU *(Bilingual Evaluation Understudy)*, welche ursprünglich für die Bewertung von maschinellen Übersetzungen entwickelt wurde. BLEU vergleicht eine generierte Zusammenfassung mit einer Referenzzusammenfassung anhand von N-Grammen (vgl. Chiusano, 2022a).

2.2.5 Textklassifikation

Implementierung

Die Umsetzung der Textklassifikation wird durch einen Prozess des maschinellen Lernens erfolgen, welcher mehrere Schritte umfasst: die Datenvorverarbeitung, die Modellentwicklung und die Modellevaluierung.

Die Vorverarbeitung von Textdaten in der Textklassifikation hat einige Ähnlichkeiten zu den Vorverarbeitungsschritten für die Textzusammenfassung, jedoch ist die Zielsetzung eine andere, weshalb angepasste Aspekte von Relevanz sind. In der in diesem Projekt betrachteten Textzusammenfassung werden die Texte zur Ermittlung der Relevanz einzelner Sätze vorverarbeitet (siehe Unterabschnitt 2.2.4), wohingegen die Texte im Rahmen der Klassifikation umgewandelt werden, um das Training eines Machine Leaarning-Modells vorzubereiten (vgl. Rani, 2021).

Im ersten Schritt ist dabei die Verteilung der Klassen innerhalb der Zielveriable der ausgewählten Daten von Relevanz, um eine Ungleichverteilung dieser für die spätere Verarbeitung zu vermeiden. Beispielsweise können hierbei Techniken wie SMOTE *(Synthetic Minority Oversampling Technique)* verwendet werden, um die Verteilung der Daten anzupassen. Liegen schon in Bezug auf die Klassen der Zielvariable gleichverteilte Daten vor, so ist ein solcher Schritt nicht notwendig. Darüber hinaus sind Schritte der Textbereinigung durchzuführen, um die Textdaten auf eine standardisierte und verarbeitbare Form zu bringen. Dies beinhaltet beispielsweise das Entfernen einzelner Zeichen, wie Interpunktionszeichen oder Sonderzeichen. Für diesen Verarbeitungsschritt kann eine einfache String-Manipulation genutzt werden. Zudem können Stoppwörter entfernt und einzelne Wörter umgewandelt werden *(z.B. Stemming, Lemmatization oder Kleinschreibung von Worten)* (vgl. Rani, 2021).

Anschließend werden die Textbestandteile in numerische Repräsentationen umgewandelt *(Vectorization)*, damit sie von den Modellen verarbeitet werden können. Dies kann beispielsweise Techniken wie tf-Idf *(Bag-of-Words)* oder Word2Vec *(Word Embedding)* beinhalten. Word2Vec ermöglicht dabei zudem die semantische Betrachtung einzelner Wörter, was einen Mehrwert bringen kann. Hierzu kann beispielsweise *„Gensim“* als Bibliothek verwendet werden (vgl. Rani, 2021).

Nach der Datenvorverarbeitung erfolgt die eigentliche Modellerstellung. Hierbei wird eine geeignete Modellstruktur ausgewählt, welche für die Klassifikationsaufgabe geeignet ist. Beispielhaft können lineare Regressions-Modelle oder Naive Bayes-Modelle erstellt werden (vgl. Rani, 2021).

Das Modelltraining ist ein entscheidender Schritt in der Entwicklung eines NLP-Modells für die Textklassifikation. Dabei werden die vorverarbeiteten Daten verwendet, um das Modell auf die spezifische Klassifikationsaufgabe zu trainieren. Das Training des Modells kann je nach Modellart variieren (vgl. Rani, 2021).

Hierbei können auch Techniken des Transfer Learning genutzt werden, um bereits vortrainierte Deep Learning-Modelle, wie zum Beispiel BERT oder GPT-2, für die Klassifikation zu verwenden, welche bereits auf großen Datenmengen trainiert wurden. Um die Modelle auf das Klassifikationsproblem anzupassen, kann hierbei eine Feinabstimmung auf den anwendungfallspezifischen Daten mit der jeweiligen Zielvariable durchgeführt werden, wobei dies auf verschiedene Weisen geschehen kann. Es kann hierbei beispielsweise die gesamte Modellarchitektur auf die spezifischen Daten abgestimmt werden. Jedoch können auch nur einzelne Schichten innerhalb der genutzten neuronalen Netze angepasst werden, wobei

die restlichen im Laufe der Feinabstimmung unverändert bleiben. Des Weiteren können zusätzliche Schichten auf Basis der Daten ergänzt werden (vgl. Prateek, 2022).

Aufgrund der Komplexität und des Umfangs des Projekts wird Transfer Learning als vielversprechende Methode für die Umsetzung bevorzugt. Allerdings kann zum gegenwärtigen Zeitpunkt nicht mit absoluter Sicherheit gesagt werden, ob die verfügbaren Ressourcen und die Rahmenbedingungen des Projekts eine erfolgreiche Implementierung von Ansätzen des Transfer Learning ermöglichen. Daher wird als Vorsichtsmaßnahme ein alternativer Ansatz in Betracht gezogen. Die endgültige Entscheidung über die verwendete Methode wird im detaillierten Projektstrukturplan festgelegt.

Datenbeschaffung

Die sorgfältige Auswahl eines geeigneten Datensatzes ist auch im Rahmen der Klassifikation von entscheidender Bedeutung für die erfolgreiche Umsetzung dieses Aspekts. Dabei ist es wichtig, einen Datensatz zu finden, der den Anforderungen des Projekts entspricht und gleichzeitig im Rahmen der verfügbaren Ressourcen liegt.

Ein für die Textklassifikation geeigneter Datensatz enthält eine Palette von Textkategorien, welche mit den entsprechenden Texten verknüpft sind. Dabei werden 4-5 unterschiedliche Oberkategorien aus dem Bereich Newsartikel anvisiert. Die Texte müssen dabei in englischer Sprache verfasst sein, um den Scope des Projekts zu treffen.

Es ist wichtig, dass der Datensatz eine ausreichende Größe aufweist, um eine repräsentative Abdeckung der verschiedenen Kategorien zu gewährleisten. Dabei ist es jedoch entscheidend, den Umfang so zu wählen, dass der Datensatz von den verfügbaren Ressourcen, einschließlich der zur Verfügung stehenden Hardware, verarbeitet werden kann. Dies bezieht sich auf die Verarbeitung auf einem handelsüblichen Computer.

Evaluierung

Bei der Evaluierung der Textklassifikation können verschiedene Metriken verwendet werden, um die Qualität der Klassifikation zu messen. Hierbei ist zu beachten, dass es sich in dem gegebenen Fall um ein Klassifikationsproblem mit mehr als zwei Klassen handelt, weshalb angepasste Metriken zu verwenden sind. Abzugrenzen ist dies von einem Fall, bei welchem es mehrere Zielvariablen zu berücksichtigen gilt (vgl. Odenthal, 2022). Um eine Klassifikation in mehr als zwei Klassen zu evaluieren, können verschiedene Metriken verwendet

werden, welche Bezug auf Metriken aus der binären Klassifikation nehmen (vgl. Varapalli, 2022). Beispielhaft sind dabei die *„Accuracy“*, welche den Anteil der korrekt klassifizierten Texte im Verhältnis zur Gesamtzahl der Texte misst, der *„Recall“*, die *„Precision“* oder der F1-Score zu nennen (vgl. Kanstren, 2020).

Sind mehrere Klassen zu berücksichtigen, so werden diese Metriken zusammengefasst für alle Klassen berechnet, wobei verschiedene Arten der Durchschnittsbildung unterschieden werden. Beispielhaft kann hierbei der Wert für jede Klasse einzeln berechnet werden, wonach der Durchschnitt über alle Klassen hinweg ermittelt wird. Auch kann die Häufigkeit der einzelnen Klassen in die Evaluierung mit einbezogen werden (vgl. Varapalli, 2022).

2.2.6 Webanwendung

Die Webanwendung des NLP-Tools wird mithilfe des Flask-Frameworks entwickelt, da dieses ein leichtgewichtiges Framework ist, welches eine einfache Entwicklung und Bereitstellung von Webanwendungen ermöglicht und verschiedene Erweiterungen bietet (vgl. Kumar, 2021).

Da Flask zudem auf der Programmiersprache Python basiert (vgl. Kumar, 2021) und die Teammitglieder über Erfahrungen mit dieser Programmiersprache wie auch mit Flask selbst verfügen, wird dieses Framework als geeignet für das Projekt angesehen. Die Ausführung des Textzusammenfassungs- und Klassifikationstools erfolgt lokal auf dem System des Benutzers, ohne die Notwendigkeit einer externen Serverinfrastruktur.

Dazu wird zunächst eine Python-Umgebung eingerichtet, um eine saubere und isolierte Entwicklungsumgebung zu gewährleisten. Dies ermöglicht die Installation der erforderlichen Abhängigkeiten, einschließlich Flask. Das Frontend der Webanwendung wird in HTML geschrieben, wobei der damit einhergehende Code in einem eigenen Ordner namens *„Templates“* festgehalten wird. Zudem wird CSS zur weiteren Gestaltung der Webanwendung verwendet.

Um die Funktionalität der Webanwendung sicherzustellen, werden das evaluierte Modell wie auch die Textzusammenfassung jeweils über eine Funktion in die Webanwendung mit eingebunden. Der Benutzer hat die Möglichkeit eine Kompressionsrate und einen Text auf der Benutzeroberfläche einzugeben, wonach diese eingelesen und an die beiden Funktionen übergeben werden. Die Resultate der durchlaufenen Funktionen werden zurückgegeben und in der Webanwendung dargestellt.

Literaturverzeichnis

Asif, M., Ali, I., Malik, M. S. A., Chaudary, M. H., Tayyaba, S., & Mahmood, M. T. (2019). Annotation of software requirements specification (srs), extractions of nonfunctional requirements, and measurement of their tradeoff. *IEEE Access*, *7*, 36164–36176.

Chiusano, F. (2022a). Learn the BLEU metric by examples. *NLPlanet*. Verfügbar 23. Mai 2023 unter https://medium.com/nlplanet/two-minutes-nlp-learn-the-bleu-metric-by-examples-df015ca73a86/

Chiusano, F. (2022b). Learn the ROUGE metric by examples. *NLPlanet*. Verfügbar 23. Mai 2023 unter https://medium.com/nlplanet/two-minutes-nlp-learn-the-rouge-metric-by-examples-f179cc285499

Gupta, V., Lehal, G. S., et al. (2009). A survey of text mining techniques and applications. *Journal of emerging technologies in web intelligence*, *1*(1), 60–76.

Kanstren, T. (2020). A look at precision, recall, and F1-score. *Towards Data Science*. Verfügbar 23. Mai 2023 unter https://medium.com/analytics-vidhya/confusion-matrix-accuracy-precision-recall-f1-score-ade299cf63cd

Korte, D., Leenen, P., & Ullrich, D. G. (2021). Fahrerlose Transportsysteme (FTS) Statusreport Technik Leitfaden Phasen, Rollen und Akteure in FTS-Projekten Edition Juni 2021.

Kumar, H. (2021). Introduction to Micro Web Framework Flask. *featurepreneur*. Verfügbar 23. Mai 2023 unter https://medium.com/featurepreneur/introduction-to-micro-web-framework-flask-78de9289270b

Kusay-Merkle, U., et al. (2018). *Agiles Projektmanagement im Berufsalltag*. Springer.

Mazumdar, S. (2023). Exploring the Extractive Method of Text Summarization. Verfügbar 23. Mai 2023 unter https://www.analyticsvidhya.com/blog/2023/03/exploring-the-extractive-method-of-text-summarization/

Mithbavkar, O., & Chauhan, A. (2021). Techniques for Extractive and Abstractive Text Summarization. Verfügbar 23. Mai 2023 unter https://medium.com/@devilchauhan0/techniques-for-extractive-and-abstractive-text-summarization-6ed44a5465f6

Odenthal, M. (2022). Multi-class vs. Multi-label Classification: What's the Difference? *MLearning.ai*. Verfügbar 23. Mai 2023 unter https://medium.com/mlearning-ai/multi-class-vs-multi-label-classification-whats-the-difference-2840a87c8dfe

Prateek, J. (2022). Transfer Learning for NLP: Fine-Tuning BERT for Text Classification. Verfügbar 23. Mai 2023 unter https://www.analyticsvidhya.com/blog/2020/07/transfer-learning-for-nlp-fine-tuning-bert-for-text-classification/

Rani, V. (2021). NLP Tutorial for Text Classification in Python. *Analytics Vidhya*. Verfügbar 23. Mai 2023 unter https://medium.com/analytics-vidhya/nlp-tutorial-for-text-classification-in-python-8f19cd17b49e

Sciforce. (2019). Towards Automatic Text Summarization: Extractive Methods. *Sciforce*. Verfügbar 23. Mai 2023 unter https://medium.com/sciforce/towards-automatic-text-summarization-extractive-methods-e8439cd54715

Varapalli, V. v. (2022). Evaluating Metrics for Multi-class Classification and Implementations. *Geek Culture*. Verfügbar 23. Mai 2023 unter https://medium.com/geekculture/evaluating-metrics-for-multi-class-classification-and-implementations-be389cb17e65

Vieweg, W. (2015). Agiles (Projekt-) Management. *Management in Komplexität und Unsicherheit: Für agile Manager*, 41–42.