**Thema: Theoretische und praktische Aspekte eines K1 geschützten Content-Filter Systems in Verbindung mit der Entwicklung und praktischen Erprobung im Unternehmensumfeld**

**Inhaltsverzeichnis**

[1. Einleitung 4](#_Toc136059190)

[**1.1.** **Einführung in das Thema und seine Bedeutung für Unternehmen** 4](#_Toc136059191)

[**1.2.** **Ziel und Lösungsweg der Arbeit** 4](#_Toc136059192)

[**1.3.** **Herkunft der Daten** 5](#_Toc136059193)

[**1.4.** **Umgebung der Arbeit** 6](#_Toc136059194)

[**1.5.** **Aufbau der Arbeit** 6](#_Toc136059195)

[2. Theoretischer Hintergrund 7](#_Toc136059196)

[**2.1.** **Grundlagen des Data-Mining** 7](#_Toc136059197)

[**2.2.** **Content-Filtering und seine Anwendungen** 7](#_Toc136059198)

[**2.3.** **Bestehende Forschung und Literaturübersicht** 7](#_Toc136059199)

[3. Methodik für die Entwicklung und praktische Erprobung des Content-Filter-Systems 8](#_Toc136059200)

[**3.1.** **Beschreibung des Forschungsdesigns** 8](#_Toc136059201)

[**3.2.** **Datensammlung und Verarbeitung** 8](#_Toc136059202)

[**3.2.1.** **URL-Bearbeitung** 8](#_Toc136059203)

[**3.2.2.** **Webseiten crawlen** 10](#_Toc136059204)

[**3.2.3.** **Datenbereinigung und -aufbereitung** 11](#_Toc136059205)

[**3.3.** **Auswahl der Trainingsdaten und Validierungsdaten** 11](#_Toc136059206)

[**3.4.** **Implementierung des Content-Filter-Systems** 11](#_Toc136059207)

[**3.4.1.** **Auswahl der Technologien und Tools** 11](#_Toc136059208)

[**3.4.2.** **Umsetzung der Algorithmen zur Klassifikation von Webseiten anhand ihrer Inhalte und URLs** 11](#_Toc136059209)

[**3.5.** **Durchführung von Experimenten und Tests** 11](#_Toc136059210)

[**3.5.1.** **Evaluation der Genauigkeit und Effektivität des Systems** 11](#_Toc136059211)

[**3.5.2.** **Überprüfung der Performance und Skalierbarkeit des Systems** 11](#_Toc136059212)

[4. Ergebnisse 12](#_Toc136059213)

[**4.1.** **Darstellung und Analyse der Ergebnisse aus den Experimenten und Tests** 12](#_Toc136059214)

[**4.2.** **Diskussion der Ergebnisse im Hinblick auf die Forschungsfragen und Hypothesen** 12](#_Toc136059215)

[**4.3.** **Interpretation der Ergebnisse im Kontext der theoretischen Grundlagen und bestehenden Forschung** 12](#_Toc136059216)

[5. Diskussion und Fazit 13](#_Toc136059217)

[**5.1.** **Zusammenfassung der wichtigsten Erkenntnisse und Ergebnisse** 13](#_Toc136059218)

[**5.2.** **Bewertung der Relevanz und Praktikabilität des Content-Filter-Systems im Unternehmensumfeld** 13](#_Toc136059219)

[**5.3.** **Empfehlungen für zukünftige Forschung** 13](#_Toc136059220)

[Bibliografische Referenzen 14](#_Toc136059221)

# Einleitung

## **Einführung in das Thema und seine Bedeutung für Unternehmen**

Mit dem Fortschritt der neuen Technologien im Bereich der Informations- und Kommunikationstechnologien (IKT) hat die Datenmenge im Internet dramatisch zugenommen und wird als Big Data bezeichnet. Big Data ist ein Datensatz, dessen Größe die Erfassungs-, Speicher-, Verwaltungs- und Analysemöglichkeiten der üblichen Datenbanksoftware übersteigt[1]. Google ist eines der Unternehmen, die sich in erheblichem Maße mit Big Data befassen, da es von Beginn seiner Entwicklung als Suchmaschine und Anbieter von Online-Diensten an mit massiven Datenmengen konfrontiert wurde. Auch ihre Infrastruktur zur Datenverarbeitung und -indexierung wurde so konzipiert, dass sie große Mengen an Informationen effizient verwalten und analysieren kann. Man kann nicht von Big Data sprechen, ohne Data Mining zu erwähnen, da man eine große Menge an Rohdaten hat, die man analysieren und in nützliche Daten umwandeln muss. „Data Mining'' auch bekannt als Knowledge Discovery in Data (KDD), ist der Prozess der Entdeckung von Mustern und anderen wertvollen Informationen aus großen Datenbeständen[2]. Dies wird in den nächsten Kapiteln näher erläutert.

Eine große Menge an Daten zu haben ist gut, aber es ist besser, sie zu sichern oder ihre Systeme vor Außenstehenden zu schützen. Dies sind die Herausforderungen, denen sich einige Unternehmen und sogar Privatpersonen heute gegenübersehen, denn eine große Menge an Informationen, die von außerhalb unseres Systems kommen, birgt das Risiko, dass unerwünschte oder schädliche Inhalte wie Spam, Datenmanipulation durch Hacker, Websites mit gewalttätigem Inhalt oder mit Inhalten, die nur für ein bestimmtes Publikum bestimmt sind, Malware oder unangemessene Inhalte in die Unternehmensumgebung gelangen.

Um diesen Herausforderungen zu begegnen, gewinnt die Entwicklung von Inhaltsfiltersystemen zunehmend an Bedeutung. Solche Systeme müssen in der Lage sein, Inhalte automatisch zu analysieren, zu kategorisieren und zu filtern, um unerwünschte Inhalte zu blockieren und den Zugang zu relevanten und sicheren Informationen zu ermöglichen.

## **Ziel und Lösungsweg der Arbeit**

Alles begann mit einem Bedürfnis oder Problem, das die Firma Telco-Tech lösen wollte. Telco-Tech ist ein innovatives, internationales und unabhängiges Unternehmen, das sich auf hochwertige Sicherheitssysteme spezialisiert hat. Das Unternehmen wollte ein System, mit dem es Webseiten kategorisieren kann, um Systeme oder Nutzer besser zu schützen. Das Ziel ist es, eine Anwendung zu entwickeln, die eine URL oder Webseite (die vom Benutzer eingegeben wird) und bis zu einem gewissen Grad auch deren Unterseiten oder verwandte Webseiten (und deren verwandte Unterseiten) klassifiziert oder kategorisiert, indem sie den Inhalt der Seite analysiert und in eine Liste vordefinierter Kategorien einordnet.

Der Lösungsweg oder das System wird wie folgt präsentiert: Das System analysiert den Inhalt der Webseite und die internen Links, die zu anderen Webseiten führen, um die richtige Kategorie zu finden. In Firewalls ist es möglich, über einen Reverse Proxy den Zugriff auf Webseiten anhand ihrer Kategorien zu verbieten. Die zu verbietenden Kategorien werden vom Administrator konfiguriert. Wenn eine Website aufgerufen wird und noch keine Kategorie im lokalen Cache gespeichert ist, fragt unser Filterserver nach der Kategorie. Für den Nutzer ist dieser Vorgang transparent, d.h. er ist nicht sichtbar.

Das überwachte Lernen wird verwendet, um dieses Problem zu lösen, weil ein Modell des maschinellen Lernens durch die Bereitstellung geeigneter Daten trainiert wird. Genauer gesagt wird ein Klassifikationsmodell verwendet. Die Ergebnisse werden anhand der Genauigkeit gemessen, die ein Maß für die Leistung eines Klassifikationsmodells beim maschinellen Lernen ist. Mit anderen Worten: Es handelt sich um den Prozentsatz der richtigen Antworten, die das Modell liefert. Die Mindestleistung, die erforderlich ist, um das Ziel dieses Projekts zu erreichen, wird zunächst über 70% betragen, und die endgültige Lösung muss dann 90% oder mehr betragen.

## **Herkunft der Daten**

Die Daten wurden für dieses Projekt bereitgestellt. Tatsächlich handelt es sich um eine Textdatei mit einer Liste von 1 185 685 Zeilen und einer Kapazität von 206,2 MB. Sie besteht aus 257 verschiedenen Kategorien, z. B. Pornografie, Shopping, Religionen, Nachrichten, Computerspiele usw. Der Zugriff auf den Inhalt einer Webseite erfolgt jedoch über eine URL.

Zunächst müsste die Datei mit den verschiedenen Domainnamen und der entsprechenden Kategorie eingelesen werden, um schließlich die Domainnamen in URLs umwandeln und das Ergebnis in einer json-Datei speichern zu können. Der nächste Schritt besteht darin, den Inhalt der Webseiten zu erhalten. Dies wird mithilfe von Bibliotheken wie BeautifulSoup für statische Seiten, die keine menschliche Präsenz benötigen, um ihren Inhalt zu erhalten, und Selenium für dynamische Seiten, die eine menschliche Präsenz benötigen, die eine Interaktion beinhaltet (ein automatischer Klick auf eine Schaltfläche), möglich sein. Das Ergebnis dieser Extraktion sollte in einer Datendatei gespeichert werden, die die folgenden Spalten enthält: Name, Kategorie, URL, auf der Seite enthaltene Wörter und Sprache. Dieser Schritt wird als "Scraping" bezeichnet.

## **Umgebung der Arbeit**

Um dieses Projekt durchzuführen, benötigt man natürlich eine ausreichend leistungsfähige Arbeitsumgebung und genügend Speicherplatz. In unserem Fall werden wir einen Desktop-Computer mit Ubuntu 22.04.1 LTS, einem i8-Prozessor mit 3,7 GHz und 1,5 TB Speicherplatz verwenden. Python 3.11 als Programmiersprache , Jupyter Notebook als Programmierumgebung und schließlich die Bibliotheken für Machine Learning und Scraping.

## **Aufbau der Arbeit**

Diese Forschungsarbeit wird in 5 Kapiteln vorgestellt.

Das erste Kapitel wird das Ziel definieren, das mit dieser Arbeit erreicht werden soll. Das zweite Kapitel behandelt einige theoretische Konzepte des Data Mining und relevante Forschungsergebnisse zu KI-basierten Inhaltsfiltersystemen. Das Forschungsdesign und die Methodik werden im dritten Kapitel beschrieben. Hier werden verschiedene Aspekte behandelt, wie die Sammlung und Verarbeitung von Daten, die Auswahl von Technologien und Werkzeugen sowie die Implementierung von Algorithmen zur Klassifizierung von Webseiten nach ihrem Inhalt. Außerdem werden Experimente und Tests durchgeführt, um die Genauigkeit und Effizienz des Systems zu bewerten. Im vierten Kapitel werden die Ergebnisse der Experimente und Tests analysiert. Anschließend werden die Genauigkeit und Effizienz des Systems sowie seine Leistung und Skalierbarkeit diskutiert und in Bezug auf die Forschungsfragen und Hypothesen interpretiert. Im letzten Kapitel werden die Ergebnisse zusammengefasst und in Bezug auf die Forschungsfragen und Hypothesen diskutiert. Außerdem werden mögliche Einschränkungen und Begrenzungen der Studie aufgezeigt und weitere Forschungsfragen für zukünftige Studien formuliert.

# Theoretischer Hintergrund

## **Grundlagen des Data-Mining**

In Kapitel 1 wurde gesagt, dass sich die Technologie drastisch weiterentwickelt und damit unsere Gewohnheiten verändert hat, aber diese Entwicklung zeichnete sich auch durch eine Fülle von Daten aus. Trotz dieser Tatsache stehen wir oft vor dem Problem der "Informationsarmut"[3], da die Daten oft in Datenbanken gespeichert werden, ohne dass sie vollständig genutzt werden, was eine fundierte Entscheidungsfindung einschränkt. Die Daten sind oft zu umfangreich und komplex, um mit herkömmlichen menschlichen Fähigkeiten verarbeitet werden zu können, weshalb es wichtig ist, leistungsfähige Werkzeuge zur Datenanalyse zu entwickeln, um wertvolles und verwertbares Wissen zu extrahieren.

Data Mining ist ein systematischer und automatisierter Prozess, mit dem große Datenmengen nach Modellen, versteckten Beziehungen und wertvollen Informationen durchsucht werden können. Sie zielt darauf ab, "Datengräber" in verwertbares Wissen umzuwandeln.[3]

Es gibt mehrere Verfahren, um Daten zu durchsuchen, nämlich: Data Mining im KDD-Prozess, Data Mining-Prozess nach SEMMA, Data Mining-Prozess nach CRISP-DM

* + 1. **Data Mining im KDD-Prozess(Knowledge Discovery in Databases)**

Der KDD-Prozess ist ein umfassenderer Prozess, der verschiedene Schritte umfasst, darunter auch das Data Mining. Data Mining gilt als ein wesentlicher Schritt im KDD-Prozess, der darauf ausgerichtet ist, Wissen aus den Daten zu extrahieren. Der KDD-Prozess wird häufig als eine Abfolge von Schritten dargestellt, einschließlich Datenauswahl, Datenaufbereitung, Data Mining, Modellbewertung und Präsentation der Ergebnisse[4].

Auf der Abbildung 2.1.1. kann man die verschiedenen Schritte des KDD erkennen:

* **Datenselektion**: Phase, in der die aus dem Datensatz ausgewählten Daten oder Datenmengen für die zu erfüllende Aufgabe geeignet und relevant sind. Es ist denkbar, sowohl interne als auch externe Informationen zu verwenden. Ein Zieldatensatz wird in einer Datei für die weitere Verarbeitung der ausgewählten Daten gespeichert. Daher liegt der Schwerpunkt auf der Mindestmenge an Daten, für die die Analyse durchgeführt werden soll.[5]
* **Datenvorverarbeitung:** In dieser Phase ist es das Ziel, einen Datensatz zu erhalten, der ausgewertet werden kann. Dazu muss die Qualität der Daten überprüft und ggf. verbessert werden. Fehlende Daten werden hinzugefügt und fehlerhafte oder widersprüchliche Daten werden korrigiert. Die bereits vorhandene Datenbank kann in dieser Phase mit weiteren Attributen angereichert werden, wenn die jeweilige Aufgabe dies erfordert. Die Kenntnis der Qualität der verfügbaren Daten ist von großer Bedeutung für die weiteren Schritte der Analyse sowie für deren Ergebnisse. Die Datenqualität hat einen erheblichen Einfluss auf die Robustheit der Analyseergebnisse[5].
* **Datentransformation**: Jetzt müssen wir sie in ein geeignetes Datenformat umwandeln, das für die Analysemethode erforderlich ist. Beispiele für solche Transformationen sind die Diskretisierung von numerischen Werten, die Umwandlung von nominalen Werten in numerische Werte oder die Gruppierung von metrischen Werten in Intervallen usw. . Das Ziel ist es, die spezifischen Anforderungen der Analysemethode für den zu bewertenden Datensatz zu erfüllen[5].
* **Data Mining:** Dies ist der zentrale Schritt des KDD zur Generierung von Wissen. Durch den Einsatz geeigneter mathematischer Methoden, die auf die jeweilige Aufgabe zugeschnitten sind, wird nun das Modell entwickelt. Das Ziel des gewählten Verfahrens ist die Suche und Entdeckung von bisher unbekannten Datenmodellen. Dazu muss die entsprechende Analysemethode gewählt werden[5].
* **Evaluation und Interpretation:** Die Ergebnisse (Datenmodelle) aus der Data-Mining-Methode werden nun ausgewertet und interpretiert. Dieser Prozessschritt erzeugt bis dahin unbekanntes Wissen. Dieses Wissen muss im Hinblick auf seine Nutzbarkeit bewertet werden. Wenn die gefundenen Muster die Aufgabe nicht in dem festgelegten Umfang erfüllen, können die vorherigen Prozessschritte wiederholt werden, um ein zufriedenstellendes Ergebnis zu erzielen[5].

**Abbildung 2.1.1.** Data Mining im KDD-Prozess[5]

* + 1. **Data Mining-Prozess nach SEMMA**(Sample, Explore, Modify, Model, Assess)

SEMMA ist auch eine andere Methode im Data-Mining-Prozess, die von einigen Datenwissenschaftlern verwendet wird, um wertvolle Informationen aus großen Informationsmengen zu gewinnen. Die verschiedenen Phasen dieser Methode sind in Abbildung 2.1.2 dargestellt. Diese Phasen müssen in chronologischer Reihenfolge durchlaufen werden, um ein Modell zu erhalten, das gute Ergebnisse liefert.

* **Sample(Datensammlung):** Zunächst werden die relevanten Daten aus dem Datensatz extrahiert. So wird ein Datensatz zusammengestellt, der alle relevanten Informationen für die weiteren Analysen enthält[5].
* **Explore(Datenprüfung)**: Die extrahierten Daten werden auf Unregelmäßigkeiten in der Datenqualität hin untersucht. Auf diese Weise wird die Datenqualität überprüft und ein besseres Verständnis der Daten erreicht[5].
* **Modify(Datenmodifizierung):** Der ausgewählte Datensatz ist für das Analyseverfahren geeignet. Durch die Datentransformation kann der Datensatz für das Analyseverfahren verwendet werden[5].
* **Model(Modellbildung):** Die Daten werden entsprechend der Analysemethode aufbereitet. Mithilfe von Software wird nach Datenkombinationen gesucht, die zuverlässige Aussagen und Antworten auf das Ausgangsproblem generieren[5].
* **Assess(Datenevaluation):** Die Ergebnisse des Analyseverfahrens werden ausgewertet. Die Nützlichkeit und Zuverlässigkeit der gewonnenen Informationen wird überprüft[5].

**Abbildung 2.1.2.** Data Mining-Prozess nach SEMMA[5]

* + 1. **Data Mining-Prozess nach CRISP-DM**

Der Data-Mining-Prozess nach CRISP-DM (Cross-Industry Standard Process for Data Mining) ist ein Referenzmodell für Data Mining, da er einen Überblick über den Lebenszyklus eines Data-Mining-Projekts bietet. Es enthält die Phasen eines Projekts, ihre jeweiligen Aufgaben und die Ergebnisse[6]. Diese verschiedenen Phasen sind in Abbildung 2.1.3 dargestellt, wobei die Reihenfolge der Phasen nicht strikt ist; der äußere Kreis in dieser Abbildung symbolisiert die zyklische Natur des Data Mining selbst. Die Pfeile zeigen nur die wichtigsten und häufigsten Abhängigkeiten zwischen den Phasen an, aber in einem bestimmten Projekt hängt es vom Ergebnis jeder Phase ab, welche Phase oder welche Aufgabe einer Phase als nächstes ausgeführt werden muss[6].

Die verschiedenen Phasen sind wie folgt:

* **Business Understanding**: Diese Anfangsphase konzentriert sich darauf, die Ziele und Anforderungen des Projekts aus der Sicht des Unternehmens zu verstehen und dieses Wissen dann in eine Definition des Data Mining-Problems und einen vorläufigen Projektplan umzuwandeln, der auf die Erreichung der Ziele ausgelegt ist[6].
* **Data Understanding:** in dieser Phase geht es darum, Daten zu sammeln, sich mit ihren Daten vertraut zu machen, Probleme mit der Datenqualität zu erkennen, Hypothesen über verborgene Informationen aufzustellen, indem man einen ersten Einblick in die Daten gewinnt[6].
* **Data Preparation**: Die Phase der Datenvorbereitung umfasst alle Aktivitäten, die darauf abzielen, aus den anfänglichen Rohdaten den endgültigen Datensatz zu erstellen. Die Aufgaben der Datenvorbereitung werden wahrscheinlich mehrmals und nicht in einer vorgeschriebenen Reihenfolge durchgeführt. Sie umfassen die Auswahl von Variablen, Datensätzen und Attributen, die Bereinigung der Daten, die Konstruktion neuer Attribute und die Transformation der Daten für Modellierungswerkzeuge[6].
* **Modeling**: In dieser Phase werden verschiedene Modellierungstechniken ausgewählt und angewendet, und ihre Parameter werden auf optimale Werte kalibriert[6].
* **Evaluation:** In dieser Phase wird zunächst die Leistung des Modells überprüft und dann festgestellt, ob es den Zielen des Unternehmens entspricht. Erst nach dieser Überprüfung kann das Modell eingesetzt werden. Es wird wichtig sein, das Modell weiter zu evaluieren und die einzelnen Schritte seiner Erstellung zu überprüfen, um sicherzustellen, dass es die Ziele des Unternehmens erreicht[6].
* **Deployment:** In dieser Phase wird ein Protokoll erstellt, das dem Kunden oder anderen Personen, die das Modell verwenden möchten, dabei hilft, zu verstehen, wie das Modell funktioniert und wie es eingesetzt werden kann. In vielen Fällen ist es der Nutzer und nicht der Datenanalyst, der die Einsatzschritte durchführt[6].

**Abbildung 2.1.3.** Data Mining-Prozess nach CRISP-DM[6]

Der Data Mining-Prozess nach CRISP-DM wird zur Analyse der Daten dieses Projekts verwendet werden.

Nach der Analyse der Daten wäre es natürlich logisch, ein Machine-Learning-Modell für das Training der Daten zu erstellen, und da die Daten in Form von Zeichenketten vorliegen, wird es sehr schwierig bis unmöglich sein, ein Modell damit zu erstellen, da Machine-Learning-Modelle Daten vom Typ numerisch erwarten. Aus diesem Grund müssen wir eine Vorverarbeitung der Daten mit einem Werkzeug der künstlichen Intelligenz namens Natural Language Processing (NLP) durchführen.

## **Natural Language Processing(NLP)**

Natural Language Processing ist ein Forschungs- und Anwendungsbereich der künstlichen Intelligenz, der die Techniken und Methoden der automatischen Verarbeitung natürlicher Sprache beschreibt. Ziel ist es, auf der Grundlage der natürlichen Sprache eine direkte Kommunikation zwischen Mensch und Computer herzustellen[7].

Natural Language Processing (NLP) versucht, die natürliche Sprache zu erfassen und mithilfe von Regeln und Algorithmen computergestützt zu verarbeiten. Dazu nutzt NLP verschiedene Methoden und Ergebnisse aus der Sprachwissenschaft und kombiniert sie mit moderner Informatik und künstlicher Intelligenz. Ziel ist es, mithilfe von Sprache eine möglichst umfassende Kommunikation zwischen Mensch und Computer zu schaffen[7].

Diese verschiedenen Methoden, die in NLP verwendet werden, um natürliche Sprache zu verarbeiten und zu analysieren, umfassen: Tokenization, Lemmatisierung, syntaktische Analyse, Erkennung von benannten Entitäten, lexikalische Disambiguierung, Sprachmodellierung, Aufbau eines Word Bag Modells, maschinelle Übersetzung, Textgenerierung, etc. Nur diejenigen, die für dieses Projekt nützlich sind, werden später vorgestellt

* + 1. **Tokenization**

Dies ist der erste Schritt der Verarbeitung natürlicher Sprache, bei der ein Eingabetext in kleinere sprachliche Einheiten, sogenannte "Tokens", unterteilt wird, wobei jede Einheit entweder ein Wort, ein Satz, ein Absatz oder etwas anderes wie eine Zahl oder ein Satzzeichen ist[8].

**Beispiel:** der Eingabetext ist: Ich habe einen Computer. Dieser Computer ist neu! Nach der Tokenization : „**Ich**“, „**habe**“, „**einen**“, „**Computer**“, „**.**“, „**Dieser**“, „**Computer**“, „**ist**“, „**neu**“, „**!**“

* + 1. **Lemmatisierung**

Dies ist ein weiterer Schritt des NLP, bei dem es sich um den Prozess handelt, die verschiedenen Formen eines Wortes (wie erschienen, erscheint) auf die kanonische oder zitierte Form des Wortes abzustimmen, die auch als Lexem oder Lemma bekannt ist (z. B. erscheinen)[9].

Im Gegensatz zum Stemming, bei dem lediglich die Suffixe entfernt werden, um die Wurzel eines Wortes zu erhalten, berücksichtigt die Lemmatisierung die Morphologie und die Bedeutung des Wortes, um seine Grundform zurückzugeben.[8].

Die Lemmatisierung wird häufig zur Normalisierung von Wörtern und zur Verringerung der Dimensionalität von Textdaten eingesetzt, was die Aufgaben der Klassifizierung, Informationssuche und Textanalyse erleichtert[10].

**Beispiel:** Wörter: habt, hat, hatte hätte. Das Lemma ist: **haben**

* + 1. **Stopwords**

Stopwords sind sehr häufige und häufig verwendete Wörter in einer bestimmten Sprache, z.B. Artikel, Pronomen, Präpositionen etc.[11]. Diese Wörter bringen in der Regel keine spezifische Bedeutung für einen Text mit sich und ihr Vorkommen in einem Text ermöglicht es nicht, diesen von anderen Texten zu unterscheiden. Daher werden sie im Rahmen der Vorverarbeitung von Textdaten häufig entfernt, um so die Konzentration auf aussagekräftigere Schlüsselwörter zu ermöglichen und gleichzeitig die Dimensionalität der Textdaten zu reduzieren[9].

* + 1. **Aufbau eines Word Bag Modells**

Nachdem alle Vorverarbeitungen wie Tokenisierung, Lemmatisierung und alle Stopwords entfernt wurden, müssen die Wörter nun in Features umgewandelt werden, das ist der Zweck dieses Schritts.

Das Words-Bag-Modell, auch bekannt als Modell zur Darstellung von Dokumenten, ist eine gängige Methode, die in der Verarbeitung natürlicher Sprache zur Darstellung und Analyse von Text verwendet wird. In diesem Modell wird ein Dokument als eine ungeordnete Menge von Wörtern dargestellt, wobei die grammatikalische Struktur und die Reihenfolge der Wörter im Text ignoriert werden. Jedes Wort wird als " Feature " betrachtet und die Anzahl der Vorkommen jedes Wortes im Dokument wird verwendet, um seinen " Wert " darzustellen[12].

die BoW(Bag of Words) wird in Form einer Matrix dargestellt, wobei jede Zeile eine Vektordarstellung ist. Um diese Matrizen zu erstellen, gibt es zwei Methoden: CountVectorizer und TF-IDF Vectorizer.

* **CountVectorizer:**

Dies ist eine Methode, die zur Vektorisierung von Textdaten verwendet wird, d. h. sie wandelt Text in numerische Daten um, die auf der Wortfrequenz basieren. Er zählt die Anzahl der Vorkommen jedes Wortes in einem Text und stellt sie in Form einer Matrix häufiger Begriffe dar[13].

**Beispiel:**

Satz1: Ich lerne Programmierung

Satz2: Peter lernt Englisch

Satz3: Die Schüler lesen die Bücher

Satz4: Das Kino ist heute geschlossen

**Nach Processing**: Tokenisierung, Lemmatisierung, Entfernung stoptword und kleine Buchstabe Umformulieren

lernen 🡪 2-mal

programmierung 🡪 1-mal

peter 🡪 1-mal

englisch 🡪 1-mal

schüler 🡪 1-mal

lesen 🡪 1-mal

bücher 🡪 1-mal

kino 🡪 1-mal

heute 🡪 1-mal

schließen 🡪 1-mal

**CountVectorizer**

**Abbildung 2.2.4.** Ablauf der Bag of words mit Countvectorizer

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | lernen | programmierung | peter | englisch | schüler | lesen | bücher | kino | heute | schließen |
| S1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| S2 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| S3 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| S4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |

Das Ergebnis ist eine 4x10-Matrix, in der jedes Token ein Feature repräsentiert und die nun in einem Machine-Learning-Modell verwendet werden kann.

* **TF-IDF Vectorizer:**

TF-IDF(Term Frequency-Inverse Document Frequency) ist ein statistisches Maß, das die Relevanz eines Wortes in Bezug auf ein Dokument in einer Sammlung bestimmt. Dieses Maß wird berechnet, indem die Anzahl der Male, die ein Wort in einem Dokument vorkommt, mit dem Kehrwert der Häufigkeit des Wortes in einer Gruppe von Dokumenten multipliziert wird[14].

TF-IDF besteht aus 02 Elementen:

- Term Frequency(**TF**): Dies ist die Anzahl, wie oft ein Wort in einem bestimmten Satz vorkommt (TF = Anzahl der Wortwiederholungen in einem Satz / Anzahl der Wörter in einem Satz)[14].

- Inverse Document Frequency(**IDF** ): ist der Logarithmus der Basis der Gesamtanzahl der Sätze geteilt durch die Anzahl der Sätze, in denen das Wort vorkommt(IDF = log( Anzahl der Sätze / Anzahl der Sätze, die das Wort enthalten))[14].

Durch Multiplikation der Frequenz von Term und IDF erhält man die Term Frequency-Inverse Document Frequency **TF-IDF = TF \* IDF**[13]**.**

TF-IDF ist von CountVectorizer zu unterscheiden und diesem überlegen, denn CountVectorizer konzentriert sich auf die am häufigsten verwendeten Wörter, TF-IDF hingegen bewertet nicht nur die Gewichtung eines Wortes im gesamten Dokument, sondern untersucht auch das Auftreten eines Wortes in einem einzelnen Dokument und als Ganzes[14].

**Beispiel:**

lernen :TF-IDF=1/2\*log(4/2)=0.151

programmierung, TF-IDF:1/3\*log(4/1)=0.2

peter, TF-IDF:1/3\*log(4/1)=0.2

englisch, TF-IDF:1/3\*log(4/1)=0.2

schüler, TF-IDF:1/3\*log(4/1)=0.2

lesen, TF-IDF=1/3\*log(4/1)=0.2

bücher, TF-IDF=1/3\*log(4/1)=0.2

kino, TF-IDF=1/3\*log(4/1)=0.2

heute, TF-IDF=1/3\*log(4/1)=0.2

schließen, TF-IDF=1/3\*log(4/1)=0.2

Satz1: Ich lerne Programmierung

Satz2: Peter lernt Englisch

Satz3: Die Schüler lesen die Bücher

Satz4: Das Kino ist heute geschlossen

**Nach Processing**: Tokenisierung, Lemmatisierung, Entfernung stoptword und kleine Buchstabe Umformulieren

**TF-IDF**

**Abbildung 2.2.4.** Ablauf der Bag of words mit TF-IDF Vectorizer

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | lernen | programmierung | peter | englisch | schüler | lesen | bücher | kino | heute | schließen |
| S1 | 0.151 | 0.2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| S2 | 0.151 | 0 | 0.2 | 0.2 | 0 | 0 | 0 | 0 | 0 | 0 |
| S3 | 0 | 0 | 0 | 0 | 0.2 | 0.2 | 0.2 | 0 | 0 | 0 |
| S4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.2 | 0.2 | 0.2 |

TF-IDF Vectorizer weist jedem Begriff eine Punktzahl zu, je nachdem, wie häufig er im Dokument vorkommt und wie wichtig er im gesamten Dokumentenkorpus ist. Begriffe, die in einem Dokument häufig vorkommen, aber in den gesamten Dokumenten selten sind, erhalten eine hohe Punktzahl, während Begriffe, die im gesamten Korpus häufig vorkommen, eine

niedrige Punktzahl erhalten. Auf diese Weise lassen sich dokumentenspezifische Schlüsselwörter hervorheben[12].

Für Natural Language Processing (NLP) gibt es 02 Bibliotheken, die in der Forschung und in der Industrie weit verbreitet sind. Dazu gehören unter anderem spaCy und NLTK (Natural Language Toolkit). Diese Bibliotheken ermöglichen es, verschiedene NLP-Aufgaben einfach durchzuführen, genaue Ergebnisse zu liefern und die Leistung zu optimieren.

## **Machine Learning**

Maschinelles Lernen ist die Wissenschaft (und Kunst), Computer so zu programmieren, dass sie aus Daten lernen können[15].

„[ Maschinelles Lernen ist die ] Disziplin, die Computern die Fähigkeit verleiht, zu lernen, ohne dass sie explizit programmiert, werden“, nach Arthur Samuel, 1959.

Mit anderen Worten: Maschinelles Lernen ist Teil der künstlichen Intelligenz und findet Anwendung bei der Erkennung von Mustern, der Entdeckung von Regeln und der Ableitung geeigneter Lösungen. Es beruht auf der Verwendung vorhandener Daten, um künstliches Wissen zu generieren, das auf Erfahrung beruht. Dieses gewonnene Wissen kann dann auf andere Probleme angewendet werden, wodurch die Aufbereitung und Nutzung unbekannter Daten angegangen werden können[16].

Der Prozess des maschinellen Lernens beginnt mit einem aufbereiteten Datensatz (d. h.: gesammelt, aufbereitet, dann bereinigt, normalisiert und in Trainings-, Test- und Validierungssätze unterteilt), in dem ein maschineller Lernalgorithmus nach Mustern und Korrelationen sucht. Sobald der Lernprozess erfolgreich abgeschlossen ist, wird das gebildete Modell zur Bewertung unbekannter Daten verwendet. Auf der Grundlage dieser Vorhersagen können dann bessere Entscheidungen getroffen werden[17].

Die Entwicklung eines Modells ist ein interaktiver Prozess, der oft mehrmals durchlaufen wird, bevor das Ergebnis eine bestimmte Qualität erreicht. In der Praxis gibt es daher immer wieder Entwicklungsschleifen, in denen ein Mensch die Ergebnisse aus dem Machine-Learning-Algorithmus bewerten muss[17].

**Abbildung 2.3.** Ablauf der Machine Learning [18]

Es gibt verschiedene Arten des maschinellen Lernens: Überwachtes Lernen, unüberwachtes Lernen, halbüberwachtes Lernen und Lernen mit Verstärkung.

* + 1. **Überwachtes Lernen**

Beim überwachten Lernen wird das Modell mit einem Satz von markierten Daten trainiert, wobei jedes Beispiel (Beobachtung) mit einer Markierung oder einer Klasse verbunden ist, was als Klassifikation bezeichnet wird. Das Modell kann auch anhand der Merkmale einer Beobachtung trainiert werden und einen numerischen Zielwert vorhersagen, was als Regression bezeichnet wird. Modell lernt aus diesen Beispielen, um die Klasse oder das Label der neuen, nicht gelabelten Daten vorherzusagen[15].

Die wichtigsten Arten von Algorithmen beim überwachten Lernen sind: K Nearest Neighbours, Lineare Regression, Logistik-Regression, Support Vector Machines, Decision Trees / Random Forests und Neuronale Netze[15].

**Abbildung 2.3.1.** Überwachtes Lernen[17]

* + 1. **unüberwachtes Lernen**

Beim unüberwachten Lernen werden die Trainingsdaten nicht beschriftet. In diesem Fall analysiert das System die Daten selbstständig und gruppiert sie nach Ähnlichkeiten (Clustering), indem es relevante Muster und Merkmale identifiziert. Eine manuelle Klassifizierung ist nicht erforderlich, aber dieser automatische Ansatz kann zu Fehlern führen. Auch bei zukünftigen Vorhersagen identifiziert das System anhand der erkannten Merkmale und Klassifizierungen, worum es sich handelt[16].

Zu den nicht überwachten Lernalgorithmen gehören die nicht überwachte Klassifizierung, das Clustering (Gruppierung), die Dimensionsreduktion usw.[17].

**Abbildung 2.3.2.** **Unüberwachtes Lernen**[17]

## **Content-Filtering und seine Anwendungen**

## **Bestehende Forschung und Literaturübersicht**

# Methodik für die Entwicklung und praktische Erprobung des Content-Filter-Systems

## **Beschreibung des Forschungsdesigns**

## **Datensammlung und Verarbeitung**

Dieser Teil wird nicht nur die Methoden behandeln, mit denen ich die Daten aus den Webseiten erhalten habe, sondern auch die Wahl der HTML-Elemente.

### **URL-Bearbeitung**

Eine URL(Uniform Resource Locator) ist eine Zeichenfolge, mit der eine Ressource im Internet, wie z. B. eine Webseite, eine Datei, ein Bild usw., lokalisiert werden kann. Sie besteht aus mehreren Teilen, darunter das Protokoll (http, https, ftp usw.), der Domainname, der Pfad und der Dateiname.

Es besteht in der Regel aus 6 Teilen, z.B. auf dieser URL: **http://www.example.com:80/path/to/myfile.html?key1=value1&key2=value2#somewhereinTheDocument**, wir werden haben:

* Scheme(**http**): er gibt das Protokoll an, das der Browser verwenden soll, um die Ressource anzufordern (ein Protokoll ist eine definierte Methode zum Austausch oder zur Übertragung von Daten über ein Computernetzwerk). Bei Websites ist das Protokoll in der Regel HTTPS oder HTTP (seine ungesicherte Version)[18].
* Authority(**www.example.com:80**): Er besteht aus dem Domainnamen (www.example.com) und dem Port (80).
  + Der Domainname gibt an, welcher Webserver angesprochen wird. Er wird oft unter einer IP-Adresse verwendet (was aber selten der Fall ist, da es viel unpraktischer ist).
  + Der Port gibt die technische "Tür" an, die für den Zugriff auf die Ressourcen des Webservers verwendet wird. Er wird normalerweise weggelassen, wenn der Webserver die Standard-Ports des HTTP-Protokolls (80 für HTTP und 443 für HTTPS) verwendet, um den Zugriff auf seine Ressourcen zu ermöglichen. Andernfalls ist er obligatorisch.[18]
* Pfad zur Ressource(**path/to/myfile.html**): es ist der Pfad zu der Ressource auf dem Webserver. In den frühen Tagen des Webs stellte ein solcher Pfad einen physischen Speicherort einer Datei auf dem Webserver dar. Heute ist er hauptsächlich eine Abstraktion, die von den Webservern verwaltet wird, ohne physische Realität.[18]
* Parameter(**key1=value1&key2=value2**): Dies handelt es sich um zusätzliche Elemente, die dem Webserver zur Verfügung gestellt werden. Diese Parameter sind eine Liste von Schlüssel/Wert-Paaren, die durch das Symbol & getrennt sind. Der Webserver kann diese Parameter verwenden, um zusätzliche Operationen durchzuführen, bevor er die Ressource zurückgibt.
* Anchor(**#somewhereinTheDocument**): es ist ein weiterer Teil der Ressource . Er stellt eine Art "Lesezeichen" innerhalb der Ressource dar und gibt dem Browser Anweisungen, wie er die Inhalte an dieser "markierten" Stelle anzeigen kann. In einem HTML-Dokument scrollt der Browser beispielsweise bis zu der Stelle, an der der Anker gesetzt ist; in einem Video- oder Audiodokument versucht der Browser, zu dem Zeitpunkt zu gelangen, den der Anker repräsentiert. Es ist zu beachten, dass der Teil nach dem #, der auch als Fragment Bezeichner bekannt ist, niemals mit der Anfrage an den Server gesendet wird.

Das Ergebnis ist, dass ein Domainname ein Teil einer URL ist, was bedeutet, dass der Domainname allein uns nicht dabei helfen kann, mithilfe von Scraping-Bibliotheken Informationen über Webseiten zu sammeln. Um dies zu erreichen, muss eine Vorverarbeitung durchgeführt werden, um Domainnamen in URLs umzuwandeln. Hier ein Beispiel für einen Domainnamen **www.heise.de**, **api.livestrong.com**, **dns-dv6irbpf1.sombrero.yahoo.net**. Um sie in URLs umzuwandeln, muss ein Protokoll (http oder htpps) oder ein Schema hinzugefügt werden. Da es sich um ein htpps-Protokoll handelt, lautet der Port 80 und 443 für https und wird automatisch festgelegt. Da der Zugriff auf eine bestimmte Ressource auf der Website in diesem Fall nicht wichtig ist, wird der Pfad zu dieser Ressource weggelassen. Dies gilt auch für die Parameter und den Anker. Das Programm, das geschrieben werden soll, funktioniert folgendermaßen, um Domainnamen in URLs umzuwandeln. Zunächst muss es jeden Domainnamen aus der Datei lesen, ihn in eine URL umwandeln (indem es das Schema hinzufügt) und dann eine Anfrage an den Server stellen, um herauszufinden, ob dieser die Anfrage des Clients, die Webseite (den Inhalt) zu liefern, verarbeiten kann. Nur diejenigen, die einen HTTP-Response-Code 200 haben, werden beibehalten und in einer JSON-Datei gespeichert. Es gibt auch andere HTTP-Antwortcodes, die die Zugänglichkeit der aufgerufenen Webseite nicht ausschließen. Am Ende wird eine kleinere Liste eindeutiger URLs mit einem HTTP-Antwortcode von 200 erhalten. Da diese Datei mehr als eine Million Zeilen enthält, wird die Verwendung mehrerer Threads erforderlich sein, um die Ausführung schneller zu machen.

### **Webseiten crawlen**

Nachdem die Domainnamen umgewandelt wurden, müssen nun die Webseiteninformationen abgerufen werden. Dies wird mithilfe eines Programms geschehen, das wir schreiben werden und das den Inhalt jedes HTML-Elements mithilfe der Beautifullsoup-Bibliotek für statische Seiten und Selenium für dynamische Seiten abrufen wird. Als HTML-Elemente werden wir die Tags Titel (der im Header) und Body-Titel, die Wörter der Meta-Beschreibung und den Body auswählen. Gemäß den Empfehlungen der Firma Telco-Tech sollten nur Webseiten mit Inhalten in Deutsch, Englisch und Französisch berücksichtigt werden. Die meisten Webseiten enthalten ein Attribut language im Header, dass die verwendete Sprache anzeigt. Es gibt jedoch eine Bibliothek namens langdetect, die es ermöglicht, die verwendete Sprache schnell zu identifizieren. Anschließend werden die gesammelten Textinformationen mit Hilfe der Spacy-Bibliothek verarbeitet, lemmatisiert (dies ist eine Technik zur automatischen Verarbeitung natürlicher Sprache, bei der die verstümmelten Formen eines Wortes normalisiert werden, um seine kanonische Form zu erhalten. Bsp: das Wort "läuft". Wenn man es lemmatisiert, erhält man das Lemma "laufen".), Stopwords entfernt (Stopwords sind Wörter, die keine ausreichende Bedeutung haben, um den Textinhalt zu verstehen. Bsp: die, das, der, zu, usw.) und alles in Kleinbuchstaben formatiert, um Unterschiede in der Groß- und Kleinschreibung zu vermeiden.

Am Ende erhält man eine CSV-Datei, die fünf Spalten enthält: eine Spalte mit dem Namen, der Kategorie, der URL, dem Wort und der Sprache. Im nächsten Schritt werden die erhaltenen Texte bearbeitet. Dazu gehört die Überprüfung auf fehlende Wörter (z. B. von Webseiten, deren Inhalte nicht korrekt abgerufen wurden), Sonderzeichen, Seiten, die vorübergehend gesperrt waren oder sich noch im Aufbau befinden, und die Entfernung von Duplikaten.

### **Datenbereinigung und -aufbereitung**

### **Auswahl der Trainingsdaten und Validierungsdaten**

## **Implementierung des Content-Filter-Systems**

### **Auswahl der Technologien und Tools**

### **Umsetzung der Algorithmen zur Klassifikation von Webseiten anhand ihrer Inhalte und URLs**

## **Durchführung von Experimenten und Tests**

### **Evaluation der Genauigkeit und Effektivität des Systems**

### **Überprüfung der Performance und Skalierbarkeit des Systems**

# Ergebnisse

## **Darstellung und Analyse der Ergebnisse aus den Experimenten und Tests**

## **Diskussion der Ergebnisse im Hinblick auf die Forschungsfragen und Hypothesen**

## **Interpretation der Ergebnisse im Kontext der theoretischen Grundlagen und bestehenden Forschung**

# Diskussion und Fazit

## **Zusammenfassung der wichtigsten Erkenntnisse und Ergebnisse**

## **Bewertung der Relevanz und Praktikabilität des Content-Filter-Systems im Unternehmensumfeld**

## **Empfehlungen für zukünftige Forschung**

# Bibliografische Referenzen

[1] „mgi\_big\_data\_exec\_summary.pdf“. Zugegriffen: 25. Mai 2023. [Online]. Verfügbar unter: https://www.mckinsey.com/~/media/mckinsey/business%20functions/mckinsey%20digital/our%20insights/big%20data%20the%20next%20frontier%20for%20innovation/mgi\_big\_data\_exec\_summary.ashx

[2] „What is Data Mining? | IBM“. https://www.ibm.com/topics/data-mining (zugegriffen 25. Mai 2023).

[3] „The-Morgan-Kaufmann-Series-in-Data-Management-Systems-Jiawei-Han-Micheline-Kamber-Jian-Pei-Data-Mining.-Concepts-and-Techniques-3rd-Edition-Morgan-Kaufmann-2011.pdf“. Zugegriffen: 12. Juni 2023. [Online]. Verfügbar unter: http://myweb.sabanciuniv.edu/rdehkharghani/files/2016/02/The-Morgan-Kaufmann-Series-in-Data-Management-Systems-Jiawei-Han-Micheline-Kamber-Jian-Pei-Data-Mining.-Concepts-and-Techniques-3rd-Edition-Morgan-Kaufmann-2011.pdf

[4] U. Fayyad, G. Piatetsky-Shapiro, und P. Smyth, „From Data Mining to Knowledge Discovery in Databases“, *AI Mag.*, Bd. 17, Nr. 3, Art. Nr. 3, März 1996, doi: 10.1609/aimag.v17i3.1230.

[5] „Grundlagen des Data Mining – Ein (Prozess-)Überblick - MoreThanDigital“, 29. November 2019. https://morethandigital.info/grundlagen-des-data-mining-ein-prozess-ueberblick/ (zugegriffen 12. Juni 2023).

[6] R. Wirth und J. Hipp, „CRISP-DM: Towards a Standard Process Model for Data Mining“.

[7] N. Litzel und S. Luber, „Was ist Natural Language Processing?“, 1. September 2016. https://www.bigdata-insider.de/was-ist-natural-language-processing-a-590102/ (zugegriffen 20. Juni 2023).

[8] C. D. Manning und H. Schiitze, „Foundations of Statistical Natural Language Processing“.

[9] S. Bird, „Natural Language Processing with Python“.

[10] „Lemmatization“. https://www.mathworks.com/discovery/lemmatization.html (zugegriffen 20. Juni 2023).

[11] K. Ganesan, „What are Stop Words?“, *Kavita Ganesan, PhD*, 19. Oktober 2014. https://kavita-ganesan.com/what-are-stop-words/ (zugegriffen 20. Juni 2023).

[12] „Jurafsky, Martin.-Speech and Language Processing\_ An Introduction to Natural Language Processing (2007).pdf“. Zugegriffen: 20. Juni 2023. [Online]. Verfügbar unter: https://pages.ucsd.edu/~bakovic/compphon/Jurafsky,%20Martin.-Speech%20and%20Language%20Processing\_%20An%20Introduction%20to%20Natural%20Language%20Processing%20(2007).pdf

[13] D. K. D. K. has years of experience as a S. D. S. H. enjoys coding, teaching, und has created this website to make M. L. accessible to everyone, „Machine Learning 101: CountVectorizer Vs TFIDFVectorizer » EML“, 6. Oktober 2022. https://enjoymachinelearning.com/blog/countvectorizer-vs-tfidfvectorizer/ (zugegriffen 21. Juni 2023).

[14] „Natural Language Processing - Understanding Count Vectorizer and TF-IDF“. https://www.linkedin.com/pulse/natural-language-processing-understanding-count-tf-idf-shankar (zugegriffen 21. Juni 2023).

[15] „Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition [Book]“. https://www.oreilly.com/library/view/hands-on-machine-learning/9781492032632/ (zugegriffen 27. Juni 2023).

[16] M. Akcay, *Machine Learning. Welche Potenziale und Hürden hat es?* GRIN Verlag, 2021.

[17] L. Wuttke, „Machine Learning: Definition, Algorithmen, Methoden und Beispiele“, *datasolut GmbH*, 24. Mai 2023. https://datasolut.com/was-ist-machine-learning/ (zugegriffen 28. Juni 2023).

[18] „What is a URL? - Learn web development | MDN“, 23. Februar 2023. https://developer.mozilla.org/en-US/docs/Learn/Common\_questions/Web\_mechanics/What\_is\_a\_URL (zugegriffen 13. Mai 2023).