MASTERARBEIT

EIN VORGEHENSMODELL ZUR ERKENNUNG UND BEWERTUNG VON ANOMALIEN AUS VERFÜGBAREN INFORMATIONEN

ausgeführt am

Studiengang

Informationstechnologien und Wirtschaftsinformatik

Von: Simon Bauer-Kieslinger

Personenkennzeichen: 2010320027

Graz, am 29. März 2022 .....

Unterschrift

EHRENWÖRTLICHE ERKLÄRUNG

Ich erkläre ehrenwörtlich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benützt und die benutzten Quellen wörtlich zitiert sowie inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

.....

Unterschrift

DANKSAGUNG

An dieser Stelle möchte ich mich bei allen bedanken, die mich während des Schreibens dieser Arbeit und meines Studiums unterstützt haben. Besonders hervorheben möchte ich meine Kommilitonen Julian Hanzlik, Florian Ortbauer, Marco Schweiger und Philipp Weihs, welche mich durch moralisch und emotional immer unterstützt haben.

Weiters möchte ich mich bei meiner Familie und vor allem bei meinem Bruder für die Unterstützung bedanken.

Besonderer Dank gilt auch meinem Betreuer Mag. (FH) Dr. rer. soc. oec. Michael Amann-Langeder für seine Unterstützung und Betreuung beim Verfassen dieser Arbeit.

KURZFASSUNG

Die Anomalie Erkennung spielt in verschiedenen Bereichen eine große Rolle. Beispielsweise versuchen Kreditkartenanbieter, betrügerische Transaktionen zu identifizieren. Registriert das System einen Einkauf im Wert von Tausenden von Euro, obwohl gewöhnlich nur Bahntickets mit dieser Karte gelöst werden, ist es sehr wahrscheinlich, dass die Karte oder die persönliche ID-Nummer gestohlen wurde. In einem anderen Szenario, wenn die Produktionslinie eines Pharmaunternehmens feststellt, dass das Endgewicht von Medikamentenkapseln 20 % höher als üblich ist, kann ein Fehler im Herstellungsprozess vorliegen. Diese Anomalie Erkennung basiert meist auf strukturierten Zahlenwerten und kann somit leicht erkannt werden.

Das Ziel dieser Arbeit ist es, Anomalien aus unstrukturierten beziehungsweise schwach strukturierten Text-Daten zu erkennen und anhand ihrer Semantik Möglichkeiten zur Interpretation und Bewertung aufzuzeigen.

Im Zuge dieser Arbeit soll herausgefunden werden, welche Datenverarbeitungsmethoden und Analysemethoden verwendet werden können, um eine bestmögliche Interpretation der Text-Daten zu erreichen. Resultierend auf diesen Erkenntnissen wird ein Vorgehensmodell erstellt. Dieses Vorgehensmodell dient als Basis für einen Prototyp, um eine automatische Auswertung erzeugen zu können.

Der Prototyp soll durch eine gezielte Auswahl von Schlüsselwörtern Anomalien in verschiedenen Themenbereichen erkennen, um so vielseitig wie möglich Verwendung zu finden.

Weiters sollen geografische Brennpunkte dieser Themenbereiche identifiziert werden und durch Analysen festgestellt werden, ob diese Brennpunkte einen positiven oder negativen Effekt mit sich bringen.

ABSTRACT

Anomaly detection plays a major role in various areas. For example, credit card providers try to identify fraudulent transactions. If the system registers a purchase worth thousands of euros, although usually only train tickets are solved with this card, it is very likely that the card or the personal ID number has been stolen. In another scenario, if a pharmaceutical company's production line detects that the final weight of drug capsules is 20% higher than usual, there may be an error in the manufacturing process. This anomaly detection is usually based on structured numerical values and thus can be easily detected.

The aim of this work is to detect anomalies from unstructured or weakly structured text data and to show possibilities for interpretation and evaluation based on their semantics.

In the course of this work, it shall be found out which data processing methods and analysis methods can be used to achieve the best possible interpretation of the text data. As a result of these findings, a procedure model will be created. This procedure model serves as a basis for a prototype, to be able to generate an automatic evaluation.

The prototype is supposed to detect anomalies in different topics by a targeted selection of keywords to be used as versatile as possible.

Furthermore, geographical hotspots of these topics are to be identified and it is to be determined by analysis whether these hotspots bring a positive or negative effect.

INHALTSVERZEICHNIS

1 EINLEITUNG

"Signale weisen immer auf etwas hin. In diesem Sinne ist ein Signal nicht eine Sache, sondern eine Beziehung. Daten werden zu nützlichem Wissen über etwas, das wichtig ist, wenn sie eine Brücke zwischen einer Frage und einer Antwort schlagen. Diese Verbindung ist das Signal."

*— **Stephen Few**, **Signal: Understanding What Matters in a World of Noise***

1.1 Vorstellung des Themas

In einer digitalen Welt wie heute, gibt es unzählige Datenquellen wie Unternehmensdaten, Nachrichten, Presseberichte bis hin zu Diskussionen in den sozialen Medien. Diese Datenquellen bestehen meistens aus unstrukturierten oder schwach strukturierten Daten. Es ist also essenziell, dass diese Daten extrahiert werden, seien es Zahlenwerte aus Unternehmensdaten oder Daten aus Textdateien. In dieser Arbeit wird ein Zusammenspiel dieser Informationsquellen erarbeitet, wobei der Fokus dieser Arbeit auf der Textdaten-Analyse steht.

Diese Textdaten beinhalten oft Informationen, welche für ein Unternehmen relevant sein können, jedoch selten ausgewertet werden. Mit statistischen und linguistischen Mitteln erschließt Text-Mining-Software aus Texten Strukturen, die die Benutzerinnen und Benutzer in die Lage versetzen sollen, Kerninformationen der verarbeiteten Texte schnell zu erkennen. Im Optimalfall liefern Text-Mining-Systeme Informationen, von denen die Benutzerinnen und Benutzer zuvor nicht wissen, ob und dass sie in den verarbeiteten Texten enthalten sind. Bei zielgerechter Anwendung sind Werkzeuge des Text-Mining außerdem in der Lage, Hypothesen zu generieren, diese zu überprüfen und schrittweise zu verfeinern.

1.2 Ziel der Masterarbeit

Das Ziel dieser Masterarbeit ist es ein Vorgehensmodell zu erstellen, welches durch Adaption in verschiedenen Bereichen eingesetzt werden kann, um Probleme oder Chancen aus unstrukturierten Daten frühzeitig zu erkennen, um entsprechend reagieren zu können. Es soll anhand eines fiktiven Beispiels folgende Frage beantwortet werden:

„Welche Möglichkeiten zur Interpretation und Bewertung ergeben sich durch die Identifikation von Anomalien und deren Semantik in Informationsflüssen?"

Durch dieses Vorgehensmodell beweist man, dass Reize beziehungsweise Anomalien im Internet erkannt werden und Aussagen für unser fiktives Beispiel getroffen werden können.

1.3 Vorgehensweise

Den Grundstein für diese Arbeit legt eine Literaturrecherche über die theoretischen Grundlagen, welche für die Beantwortung der Forschungsfrage essenziell sind. Durch die erarbeiteten Erkenntnisse der theoretischen Grundlagen, wird ein Vorgehensmodell erarbeitet. Die Aufgabe des Vorgehensmodells ist es, die auftretenden Aufgabenstellungen in einer logischen Ordnung darzustellen. Das Vorgehensmodell ist ein organisatorisches Hilfsmittel und eine Schritt für Schritt Anleitung um die Forschungsfrage bestmöglich zu beantworten.

Auf Basis des Vorgehensmodells wird ein Prototyp in der Programmiersprache „R“ entwickelt, um eine automatische Durchführung dieses Prozesses auszuführen. Dieser Prototyp wird dann in einem Fallbeispiel eingesetzt, um die Validität des Vorgehensmodell zu beweisen.

Anschließend werden die Ergebnisse dieser Arbeit zusammengefasst und mögliche Erweiterungs- beziehungsweise Verbesserungsmöglichkeiten aufgezeigt.

1.4 Anomalie Erkennung

In der heutigen Industrie muss täglich sichergestellt werden, dass die Produktionslinie ordnungsgemäß läuft oder das Produkt von guter Qualität ist. Diese Prozesse werden auf unterschiedliche Weise getestet, seien es optische oder physikalische Sensoren. Zum Beispiel würde ein Schreiner / eine Schreinerin die neu gefertigten Stühle auf Schleiffehler prüfen, und ein Hersteller / eine Herstellerin von Elektronikkabeln würde prüfen, ob das Kabeln den Strom gut leitet. In jedem Berufsfeld wird nach möglichen Anomalien gesucht. Daher ist die korrekte Erkennung solcher Anomalien unerlässlich, um die Produktqualität sicherzustellen und die Produktion zu optimieren.

In einer computerisierten und vernetzten Welt hat man es mit der Datenverarbeitung und deren Analyse verbundenen Problemansatz zu tun. Beim Schreiner / der Schreinerin geht es darum, anhand von Fotos des Stuhles die Fehler zu entdecken. Beim Kabelhersteller werden die elektrischen Sensordaten analysiert.

Die automatische Erkennung von Anomalien in Datensätzen ist eine komplexe Aufgabe, die Bereiche wie maschinelles Lernen, Statistik und Data Mining umfasst. Die Art der Daten, die verfügbaren Informationen, die Art der Anomalien und die erwarteten Ergebnisse bestimmen die Wahl des zu verwendenden Algorithmus (Mehrotra, Mohan, & Huang, 2017).

Praktisch ausgedrückt geht es darum, festzustellen, welche Werte in einem Datensatz problematisch sind. Beispielsweise versuchen Kreditkartenanbieter, betrügerische

Transaktionen zu identifizieren. Registriert das System einen Einkauf im Wert von Tausenden von Euro, obwohl gewöhnlich nur Bahntickets mit dieser Karte gelöst werden, ist es sehr wahrscheinlich, dass die Karte oder die persönliche ID-Nummer gestohlen wurde. In einem anderen Szenario, wenn die Produktionslinie eines Pharmaunternehmens feststellt, dass das Endgewicht von Medikamentenkapseln 20 % höher als üblich ist, kann ein Fehler im Herstellungsprozess vorliegen.

Es ist also wichtig Anomalien zu erkennen. In der Anomalie Erkennung von Textdaten kommt noch ein weiterer Faktor zur Geltung. Die Semantik. Die Semantik ist ein Teilgebiet der Linguistik, das sich mit den Bedeutungen sprachlicher Zeichen und Zeichenfolgen befasst. Durch Einsatz richtiger Methoden, kann somit eine erkannte Anomalie bewertet werden.

2 THEORETISCHER RAHMEN

Der Begriff Data Mining und Datenanalyse ist in einer digitalen Welt wie sie heutzutage vorzufinden ist, nicht mehr weg zu denken. Die meisten Datenbeschaffungsprozesse und Datenanalysen werden hauptsächlich auf strukturierte Daten bezogen, also Zahlenwerte, welche gut verarbeitbar sind. Da es sich in dieser Arbeit um Textdaten, also unstrukturierte Daten handelt, kommt ein neues Teilgebiet des Data Mining und der Datenanalyse zum Einsatz. Das Text Mining und die Textanalyse.

Der theoretische Rahmen befasst sich mit vorwiegend mit diesen zwei Themengebieten, da sie für das Vorgehensmodell essenziell sind.

Wie bereits in Kapitel 1.2 soll das Vorgehensmodell auf verschiedene Bereiche beziehungsweise Projekte angewendet werden können. Das Vorgehensmodell beruht auf unstrukturierten oder schwachstrukturierten Textdaten. Um Bedeutungsstrukturen dieser Daten zu entdecken, kommen Algorithmus-basierte Analyseverfahren zum Einsatz (Zong, Xia, & Zhang, 2021).

In diesem Kapitel werden die Themengebiete generisch beschrieben und welche Technologien verwendet werden. In einem kommenden Kapitel wird ein Vorgehensmodell entwickelt. Dieses Vorgehensmodell wird mittels eines Fallbeispiels auf ein Projekt angewandt und evaluiert.

Die Theorie des Vorgehensmodell basiert auf dem allgemeinen Prozess von Ted Kwartler im Buch „Text Mining in Practice with R“ für Textmining.

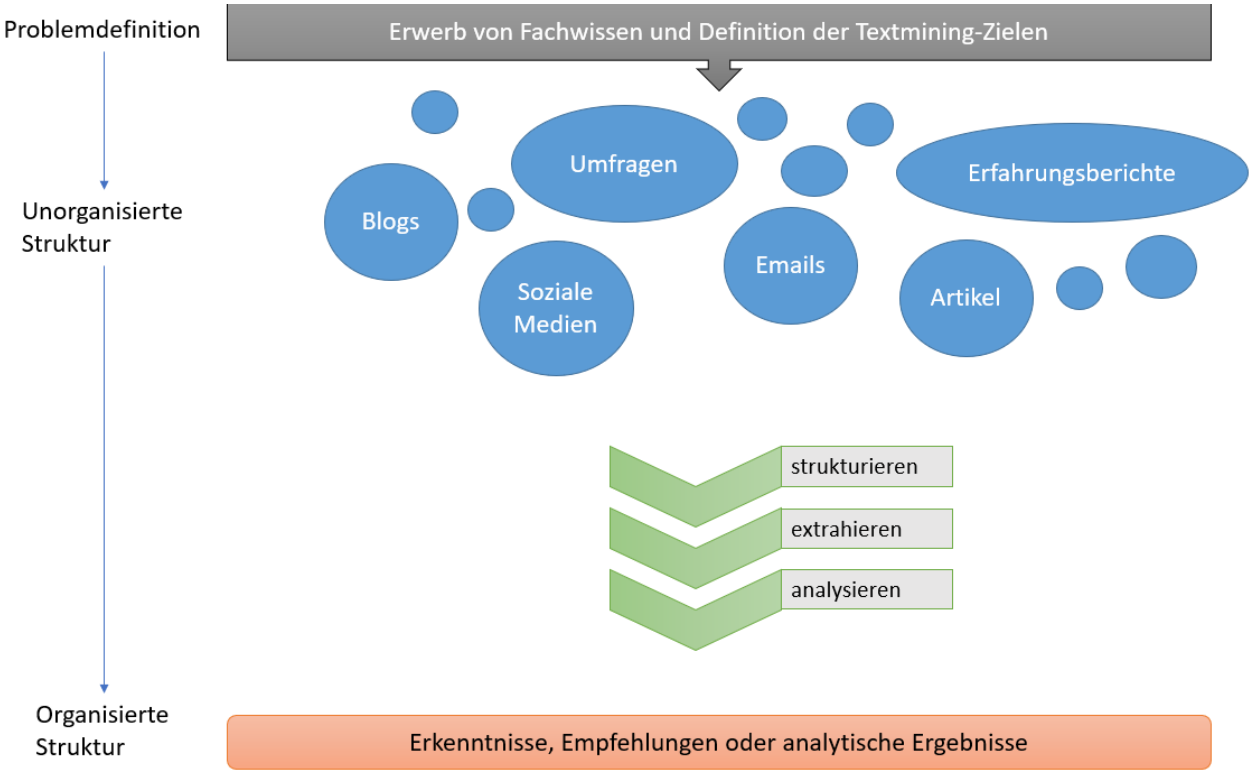


Abbildung 2-1: Allgemeiner Textmining Prozess in Anlehnung an (Kwartler, 2017)

2.1 Komponenten des Textmining

Zum besseren Verständnis und zur Strukturierung des Prozesses wird das System analysiert und mittels Diagramms in einer Art Bausteinsicht dargestellt.

2.1.1 Komponenten Überblick

Abbildung 2-2 zeigt die Abgrenzungen des Kontext beziehungsweise des System in „Input → Verarbeitung → Output“.

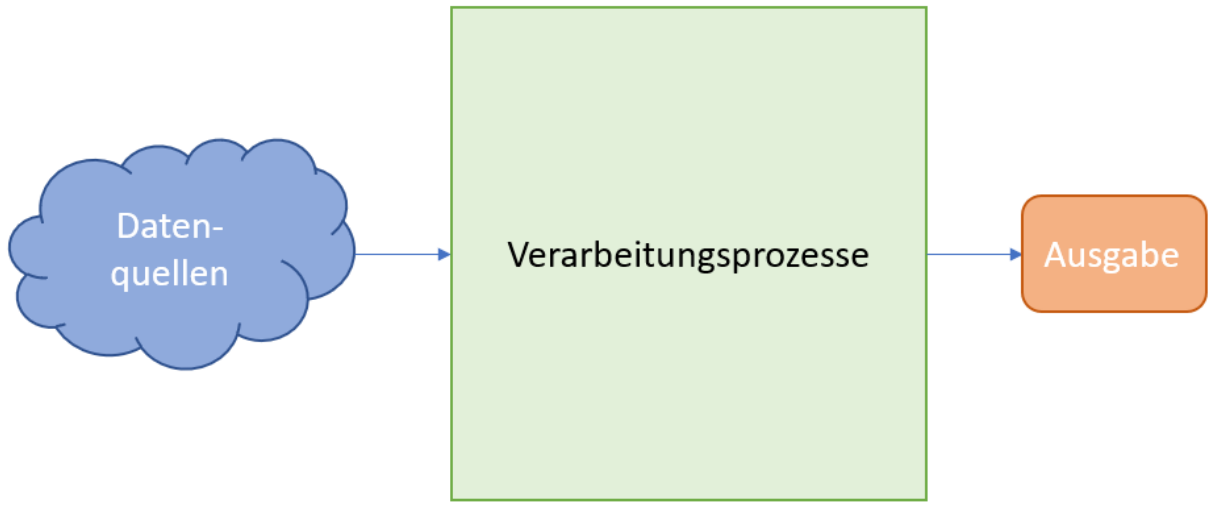


Abbildung 2-2: Textmining Komponenten Überblick (Quelle: eigene Darstellung)

- **Datenquellen**
  Es dienen verschiedenste Quellen zur Datenbeschaffung beziehungsweise als Input für die Verarbeitungsprozess. Unter Daten in der Informatik versteht man generell Daten, welche von Maschinen lesbar und bearbeitbar sind und deren Informationen digital darstellen darstellbar sind. (Schmuller, 2017) Wie bereits in der Einleitung beschrieben, fokussiert sich die Arbeit auf Textdaten. Aus dem Internet können Daten von Nachrichtensendern wie zum Beispiel „orf.at“ genutzt werden. Weiters können Beiträge auf sozialen Plattformen wie „reddit.com“ oder Diskussionen von Beiträgen auf „twitter.com“ als Datenquelle dienen. Außerdem können auch unternehmensinterne Quellen für die Datengenerierung dienen wie zum Beispiel Reviews, Umfragen, Emails, ERP-Systeme, Knowledge-Base oder Prozesse.
- **Verarbeitungsprozesse**
  Die Verarbeitungsprozesse wandeln die Daten der Datenquellen in eine Datenstruktur um, welche für Analysen benötigt werden. Im folgenden Kapitel werden die Verarbeitungsprozesse genauer betrachtet.
- **Ausgabe**
  Die Ausgabe ist die Summe der Informationen, welche in den Verarbeitungsprozessen generiert wird.

2.1.2 Überblick Verarbeitungsprozess

Um die Funktionen näher zu betrachten, wird in diesem Kapitel der Fokus auf die Verarbeitungsprozesse und dessen Komponenten gelegt.

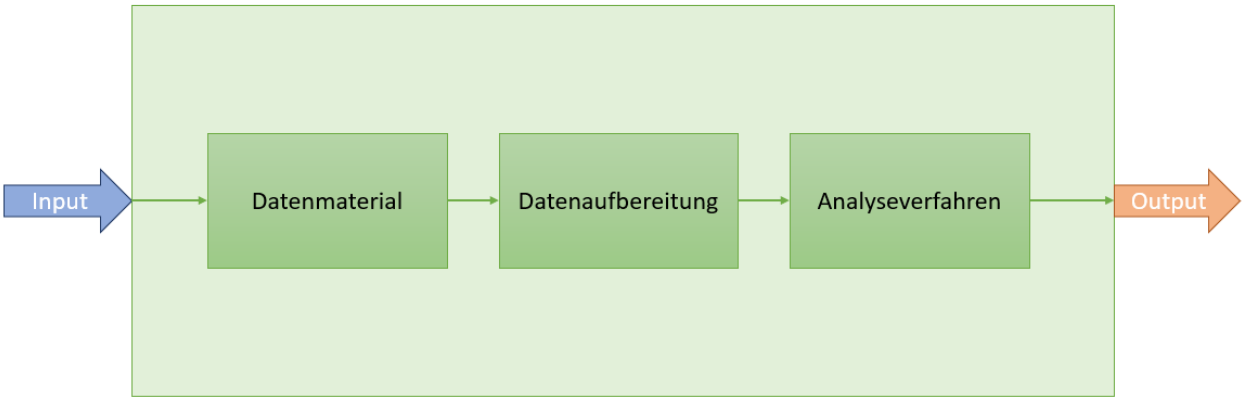


Abbildung 2-3: Überblick Verarbeitungsprozess (Quelle: eigene Darstellung)

Der Fokus des Verarbeitungsprozesses besteht aus drei Hauptkomponenten. Diese bestehen aus dem Datenmaterial, der Datenaufbereitung und dem Analyseverfahren.

- **Datenmaterial**
  Das Datenmaterial stammt aus diversen Quellen wie dem Internet und kommt oft in Form von Textdaten vor, deshalb müssen Verfahren angewendet werden, um Daten aus den Textdaten zu bekommen. Das Web besteht überwiegend aus unstrukturierten Texten. Um relevante Informationen aus dieser Unmenge von Textdaten zu sammeln, bedient man sich der Methode „Web-Scraping“. Innerhalb des unstrukturierten Textes ist man oft an bestimmten Informationen interessiert, vor allem, wenn man die Daten mit quantitativen Methoden analysieren will. (Kumar & Paul, 2016)

- **Datenaufbereitung**
  Aufgrund der unterschiedlichsten Strukturierungen der Textdaten, werden die Daten in dieser Komponente vereinheitlicht, dass alle Datensätze dieselbe Struktur haben. Ein sehr weit verbreitetes Format ist XML (Extensible Markup Language) zur Darstellung hierarchisch strukturierter Daten (Weiss, Nitin, Zhang, & Damerau, 2005). Für das Vorgehensmodell dieser Arbeit, werden die Daten jedoch in einem Dataframe (tabellarische Form) strukturiert da die Programmiersprache R diese Struktur für weitere Analysen benötigt.

- **Analyseverfahren**
  Diese strukturierten Daten werden durch diverse Analyseverfahren verarbeitet und analysiert. Auf Basis dieser Analyse wird dann der Output generiert. Bei den Analyseverfahren werden vor allem Werkzeuge der deskriptiven Statistik eingesetzt.

2.2 Datenmaterial

Wie bereits erwähnt, werden Daten aus verschiedenen Quellen verwendet. Es können Daten aus Mitarbeiterinnen- und Mitarbeitergesprächen, Umfragen, Kundinnen- und Kundenfeedbacks, Emails oder beliebige Daten aus dem Internet sein. Jegliche Daten müssen in digitalisierter Form vorliegen, um verwendet werden zu können. So müssen Umfragen, Mitarbeiterinnen- und Mitarbeitergespräche und Kundinnen- und Kundenfeedbacks falls nötig in eine textuelle digitale Form gebracht werden.

2.2.1 Web-Scraping

Bei Online-Daten kommt, die Methode „Web-Scraping“ zum Einsatz. Web-Scraping beschreibt das automatische Auslesen und Extrahieren von Inhalten einer Website. Wenn man Web-Scraping betreibt, muss man die Teile des Dokuments identifizieren und extrahieren, welche relevante Informationen enthalten. (Munzert, Rubba, Meißner, & Nyhuis, 2015)

Im Idealfall kann man Webseiten mit XPath[1] Operationen auslesen, aber oft sind relevante Informationen über ein HTML-Dokument verstreut. Um diese Informationen zu erlangen, kann man reguläre Ausdrücke[2] (regular expression, Regex) verwenden. Diese regulären Ausdrücke bieten eine Syntax für systematischen Zugriff auf Muster im Text (Kumar & Paul, 2016).

In den nächsten Abbildungen wird eine Informationsgewinnung mit Xpath und mit regulären Ausdrücken anhand eines Onlineartikels gezeigt.

Abbildung 2-4: Online Artikel (Quelle "orf.at")

In dem folgenden Bild sieht man den Quellcode der Abbildung 2-4.

[1] Die XML Path Language ist eine Abfragesprache, um Teile eines XML-Dokumentes zu adressieren und auszuwerten.

[2] Ein regulärer Ausdruck ist eine Zeichenfolge, die eine Kombination aus
normalen Zeichen und speziellen Meta-Zeichen oder Meta-Sequenzen besteht.
Metazeichen und Metasequenzen sind Zeichen oder Abfolgen von
von Zeichen, Mengen, Orte oder Arten von Zeichen darstellen (Stubblebine, 2007).

```
▼<div class="story-lead">
   <h1 class="story-lead-headline">Leere Tankstellen in GB: Soldaten helfen ab
   Montag aus</h1>
  ▶<div class="story-meta">…</div> flex
  </div>
▼<div id="ss-storyContent" class="story-content">
  ▼<div class="story-story">
    ▼<p> == $0
       "Die britische Regierung setzt ab Montag die Armee zur Überbrückung der
       Engpässe bei der Benzinversorgung ein. 200 Soldaten – davon 100 Fahrer –
       würden ein entsprechendes Training am Wochenende beenden und könnten dann
       ab Montag mit Lieferfahrten starten, teilte die Regierung gestern mit."
     </p>
```

Abbildung 2-5: Quellcode des online Artikels (Quelle "orf.at")

Durch die XPath Abfragesprache, kann man auf verschiedene Teile des Artikels zugreifen. In der folgenden Abbildung sind die XML-Elemente des Quellcodes aus Abbildung 2-5 für ein besseres Verständnis angeführt.

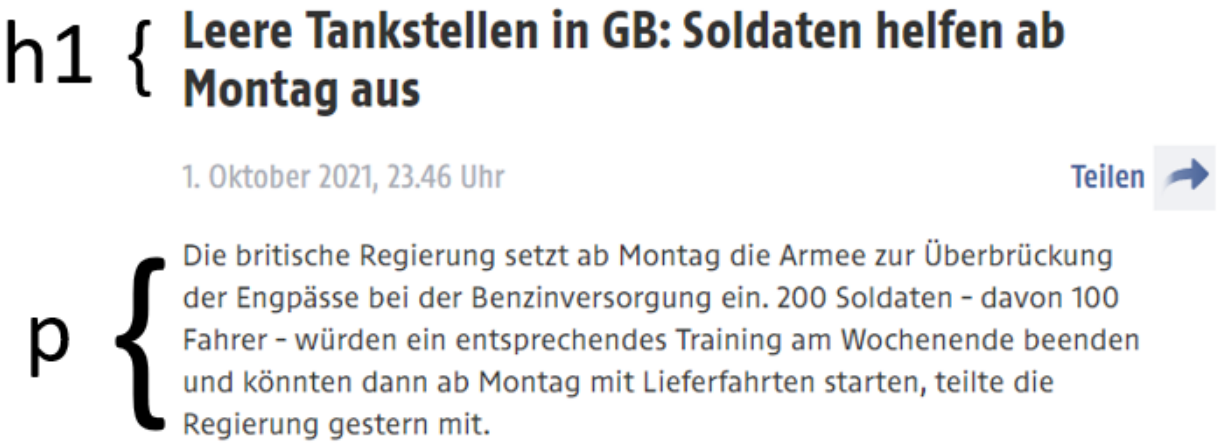


Abbildung 2-6: Artikel + XML Elemente (Quelle: „orf.at“ und eigene Darstellung)

Für die Datengenerierung aus online Quellen wird das R-Paket „rvest“ verwendet. Mit Hilfe der URL kann dieses Paket mit der Funktion read_html(Pfad) auf den Quelltext zugreifen. Mit dem Befehl html_nodes(„Element“) greift man auf jedes Element der Seite zu. Gibt es mehr als eine „h1“ Überschrift auf dieser Website, kommen mehrere Ausgaben zurück. Dies ist für das Extrahieren von Paragrafen sehr vorteilhaft. Der Code html_text() gibt den Text aus, der Zwischen den Tags (<h1>Text zum Auslesen</h1>) steht. (Kwartler, 2017)

Für die regulären Ausdrücke ändert sich der Code nur ein wenig. Es wird im html_nodes() nicht ein Element eingegeben, sondern ein regulärer Ausdruck.

Ein weiterer wichtiger Teil der Datengenerierung ist die Extraktion von Zahlen aus Strings. Man nehme an, der Artikel aus Abbildung 2-4 ist immer gleich, das heißt, gleicher Aufbau und gleicher Text. Das Einzige was sich ändert, ist der Wert der Soldaten. An einem Tag sind es 150, am nächsten 300 und am folgenden 200. Somit ist es bedeutend diese Werte zu extrahieren.

Durch dieses „rvest“ Paket hat man somit einen weiten Bereich der Datengenerierung von online Informationsquellen abgedeckt. Die generierten Daten werden in dem Dateiformat CSV (Comma-separated values) gespeichert, um für die Datenaufbereitung verwendet werden zu können.

Eine Alternative zu „rvest“ ist das „RSelenium“ Paket. Der größte Unterschied dieser Pakete kommt bei dynamischen Webseiten zu tragen. „rvest“ ist ein performantes Paket für statische Webseiten, welche sogar Interaktionen mit Passwort geschützten Sessions zulässt, jedoch stößt das Paket bei dynamischen Seiten an seine Grenzen. Falls in dem Vorgehensmodell eine dynamische Seite für die Datengenerierung ausgelesen und extrahiert werden soll, muss somit das „RSelenium“ Paket zu Hilfe gezogen werden. Vorerst ist eine Datenbeschaffung einer dynamischen Webseite jedoch nicht geplant, somit wird nicht weiter auf das „RSelenium“ Paket eingegangen.

Der entscheidende Faktor für die Entscheidung des „rvest“ Paketes beruht aber auf der Performance. Das „RSelenium“ Paket öffnet einen eigenen Chrome Browser und „spielt“ das Szenario der Datengenerierung im Browser ab. Es können mit der Programmiersprache R Mausklicks und Tastatureingaben an den Browser gesendet werden, um die Probleme der asynchronen Webseiten zu umgehen, was aber eine lange Wartezeit der Datengenerierung mit sich bringt (Munzert, Rubba, Meißner, & Nyhuis, 2015). Wohingegen das „rvest“ Paket keinen eigenen Browser öffnen muss, da die Internetadresse als Eingabeparameter genügt.

2.2.2 Bots

Bots sind Softwareprogramme, die Befehle ausführen, auf Nachrichten antworten oder Routineaufgaben wie Online-Suchen durchführen können, entweder automatisch oder mit minimalem menschlichem Eingriff. Das kann von Chat-Bots mit automatischem Kundenservice bis zu social-Media Bots gehen. Diese social Media Bots werden häufig in sozialen Netzwerken wie Twitter, Instagram oder Reddit eingesetzt. Diese Bots werden meist für Propaganda Zwecke eingesetzt (Sachs-Hombach & Zywietz, 2018).

Da diese Daten von social Bots die Anomalie Erkennung beeinflussen kann, muss im Vorgehensmodell genau auf die verwendeten Daten geachtet werden.

2.3 Datenaufbereitung

Die Datenaufbereitung beginnt vorzugsweise beim Datenmaterial / der Datengenerierung. Web-Scraping bietet den Vorteil, dass Daten vorselektiert werden können. Das resultiert in einem geringeren Zeitaufwand in der Datenaufbereitung. Wie im Kapitel oberhalb beschrieben, versucht man nur die relevanten Informationen zu generieren. Beim Auslesen von ganzen Webseiten kann der Prozess der Vorselektion jedoch mehr Zeit in Anspruch nehmen als das Aufbereiten der Daten.

Wie bereits in Kapitel 2.1.2 erwähnt, wird in der Datenaufbereitung die Vereinheitlichung der verschiedensten Datenquellen erlangt. Hierfür müssen alle Daten in ein vorgegebenes, für die

Datenanalyse vorbereitetes, Dataframe[3] gebracht werden. Dies kann durch Hinzufügen, Verschieben oder Entfernen von Spalten und Reihen gemacht werden (Wickham & Grolemund, R for Data Science, 2017).

Nachdem man die Daten strukturiert hat, kann man sich entscheiden mit welchem R-Paket man weiterarbeiten will. Es gibt mehrere Pakete, welche sich mit Textmining und Textanalyse auseinandersetzen. Eines der beliebtesten Textmining Pakte ist „tidytext“. Das Paket „tidytext“ harmoniert sehr gut mit einem für statistische Auswertungen verwendetem Paket namens „tidyverse“. „tidyverse“ ist eines der am häufigsten verwendeten Pakete bei der R Programmierung (Wickham & Grolemund, R for Data Science, 2017).

Ein weiteres Paket welches sich mit der qualitativen Analyse von Textdaten beschäftigt ist „quanteda“. Es bietet viele Funktionen und benötigt für einige Analysen keine weiteren Pakete. „quanteda“ ist ein R-Paket, das ein umfassendes Toolkit für die Verarbeitung natürlicher Sprachverarbeitungsaufgaben wie Korpusmanagement, Tokenisierung, Analyse und Visualisierung bietet. Es verfügt über umfangreiche Funktionen zur Anwendung von Wörterbuchanalysen, zur Untersuchung von Texten mit Hilfe von Schlüsselwörtern im Kontext, zur Berechnung von Dokument- und Merkmalsähnlichkeiten und zur Entdeckung von Mehrwortausdrücke durch Kookkurrenzen. (Benoit, et al., 2018)

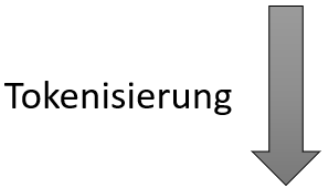
Der Vorgang in der weiteren Datenaufbereitung ist in beiden Paketen sehr ähnlich, weshalb das weitere Vorgehen nur mit „tidytext“ beschrieben wird.

Da eine Analyse mit diesen Textdaten in dieser Form nur schwer durchzuführen ist, muss der Inhalt „tokenisiert“ werden. Tokenisierung ist der Prozess der Zerlegung eines Textstroms, einer Zeichenfolge oder eines definierten Dokuments in Phrasen, Wörter, Symbole oder andere sinnvolle Elemente, die Token genannt werden. Das Ziel der Tokenisierung ist die Erkundung der Wörter in einem Satz. Bevor man den Text mit einem Sprachprozessor analysiert, muss man die Wörter normalisieren. Die Tokenisierung liefert verschiedene Arten von Informationen über den Text, wie zum Beispiel die Anzahl der Wörter oder Tokens in einem Text (Kumar & Paul, 2016).

Die folgenden Abbildungen zeigen, was bei einer Tokenisierung mit einem Dataframe passiert.

[3] Ein Dataframe beinhaltet eine geordnete Sammlung von Spalten. Jede Spalte besteht aus einem eindeutigen Daten-Typen, aber verschiedene Spalten haben verschiedene Typen.

Titel	Inhalt	Datum
Titel eines Artikels	Inhalt eines Artikels	Datum



Titel	Wort	Datum
Titel eines Artikels	inhalt	Datum
Titel eines Artikels	eines	Datum
Titel eines Artikels	artikels	Datum

Abbildung 2-7:Tokenisierungsprozess (Quelle: eigene Abbildung in Anlehnung an (Kwartler, 2017))

Durch die Zerlegung der Spalte „Inhalt“ entsteht aus einer Zeile mit drei Wörtern, drei Zeilen mit einem Wort.

Eine Tokenisierung bedeutet nicht, dass jeder Satz in einzelne Wörter geteilt wird. Tokens können eine unterschiedliche Anzahl an Wörtern beinhalten. Es gibt Textanalysen, welche eine Ein-Wort-Tokenisierung benötigen und Analysen, welche eine Zwei-Wort oder eine Satz-Tokenisierung benötigen.

Die Untersuchung des tf-idf[4] (term frequency times inverse document frequency) von Bigrammen (Zwei-Wort-Tokenisierung) anstelle einzelner Wörter hat Vor- und Nachteile. Paare von aufeinanderfolgenden Wörtern können eine Struktur erfassen, die nicht vorhanden ist, wenn man nur einzelne Wörter zählt, und somit einen Kontext liefern, der Token verständlicher macht (Silge & Robinson, 2017). Zum Beispiel ist "3 G" oder „G Nachweis“ informativer als "G" alleine.

2.3.1 Stoppwörter

Ein wichtiger Aspekt, der zum Teil zur Datenaufbereitung gehört ist das Entfernen der Stoppwörter. Stoppwörter sind Wörter in der Informationsrückgewinnung, die meist sehr häufig auftreten, aber gewöhnlich keine Relevanz für die Erfassung des Dokumenteninhalts besitzen (Weiss, Nitin, Zhang, & Damerau, 2005). Stoppwörtern in der deutschen Sprache können bestimmte oder unbestimmte Artikel sein, Konjunktionen und häufig gebrauchte Präpositionen.

[4] eine Größe zur Kennzeichnung von Begriffen, die für ein bestimmtes Dokument besonders wichtig sind

Die folgende Abbildung gibt einen kurzen Einblick auf den Einfluss der Stoppwörter. Es wurden einmal die Wörter ohne Stoppwörter gezählt und einmal mit. Es gibt mehrere Pakete, welche Stoppwörter für verschiedene Sprachen haben. Da das R-Paket „stopwords“ verschiedene Sprachen umfasst und das Vorgehensmodell Sprachunabhängig aufgebaut wird, wird dieses Paket verwendet. Um die Stoppwörter aus einem tokenisierten Datensatz zu bekommen, müssen die Wörter des Datensatzes mit den Wörtern der „stopwords“ Bibliothek verglichen und entfernt werden.

	words_without_stopwords	n	words_with_stopwords	n
1:	mehr	57	die	624
2:	prozent	46	der	579
3:	eu	39	und	411
4:	sagte	39	in	396
5:	sei	39	den	203
6:	heute	37	mit	188
7:	österreich	35	für	170
8:	gestern	32	von	168
9:	jahr	31	das	167
10:	menschen	31	zu	167

Abbildung 2-8: Vergleich Wörter mit und ohne Stoppwörter (Quelle: Eigene Darstellung)

Man sieht einen deutlichen Unterschied von am häufigsten vorkommenden Worten. In der Analyse würden mit den Stoppwörtern die relevanten Wörter einen zu kleinen Prozentsatz ausmachen, dass sie detektiert werden können.

2.4 Analyseverfahren

In diesem Kapitel werden verschiedene Analyseverfahren aufgezeigt, welche für das Vorgehensmodell relevant sein können. Kernoperationen der meisten Verfahren sind dabei die Identifizierung von Verteilungen, Mengen und Abhängigkeiten.

Ein wichtiger Faktor, dass die relevanten Daten extrahiert werden, sind sogenannte Schlüsselwörter. Man kann diese Schlüsselwörter in einem Text erkennen und verarbeiten. Als Beispiel hierfür kann eine Zeitreihen Analyse gemacht werden. (Feldman & Sanger, 2007)

2.4.1 Zeitreihenanalyse

Eine Zeitreihenanalyse wird aus einer Sequenz von Datenpunkten gebildet. Diese Datenpunkte werden immer zur gleichen Zeit erfasst (pro Sekunde, pro Tag, etc.). Weiters befasst sich die Zeitreihenanalyse mit der Vorhersage von Trends zu ihrer künftigen Entwicklung. (Kirchgässner & Wolters, 2007)

Das kommende Beispiel zeigt das Aufkommen eines Schlüsselwortes pro Tag. In diesem Fall ist das Schlüsselwort „Corona“ und es wird in allen Artikeln dieses Tages nach diesem Schlüsselwort gesucht und summiert.

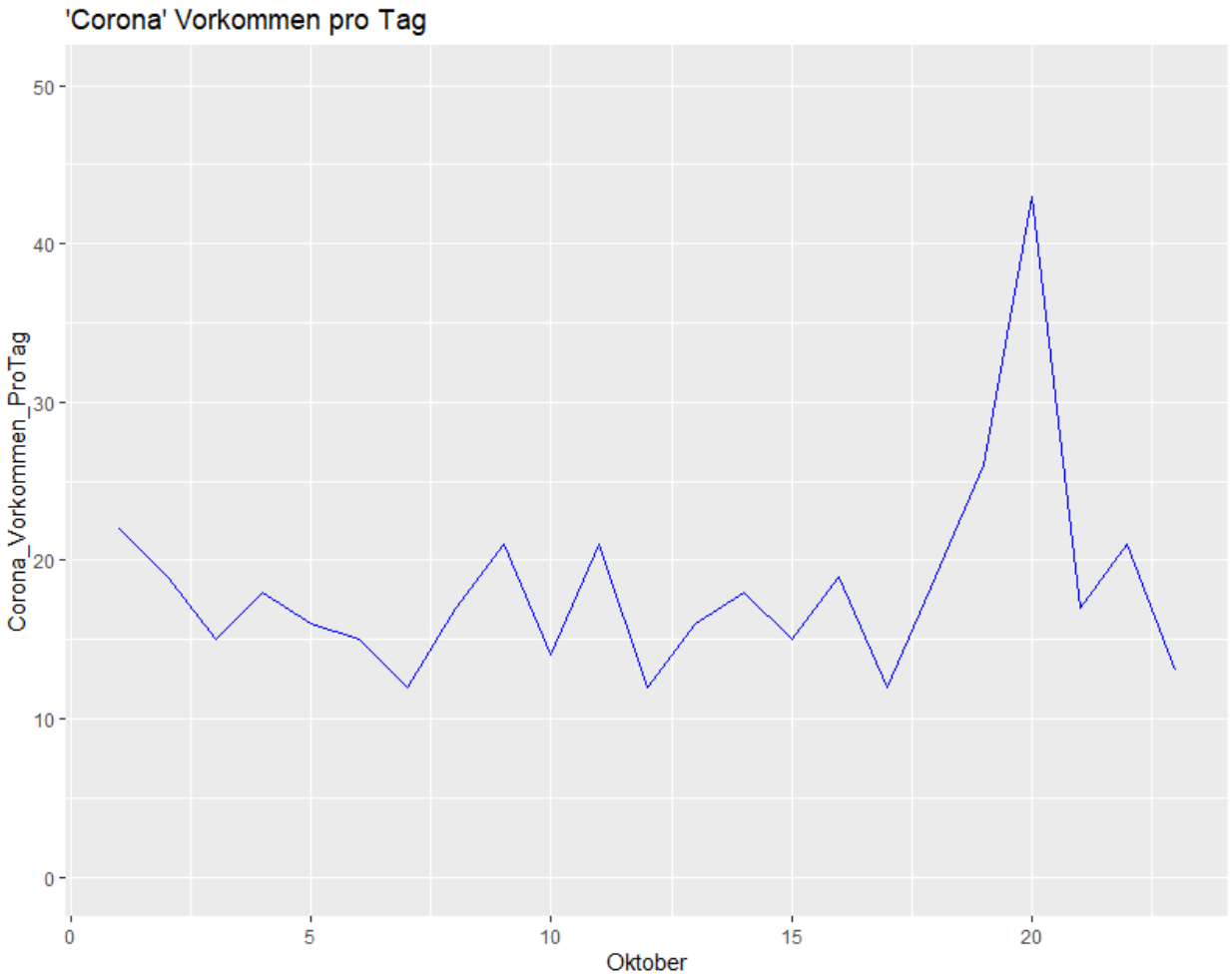


Abbildung 2-9: Schlüsselwort 'Corona' Vorkommen pro Tag (Quelle: eigene Abbildung)

Man erkennt in diesem Diagramm, dass am 20. Oktober eine erhöhte Anzahl des Schlüsselwortes vorgekommen ist. Dieser Ausschlag kann als Reiz oder Anomalie erkannt werden. Zur Evaluierung dieses Diagramms, können zum Beispiel Daten der neuer Corona Fälle hinzugezogen werden. Dieser Vergleich dient zur Evaluierung, ob an diesem Tag ein berechtigter Reiz erkannt worden ist.

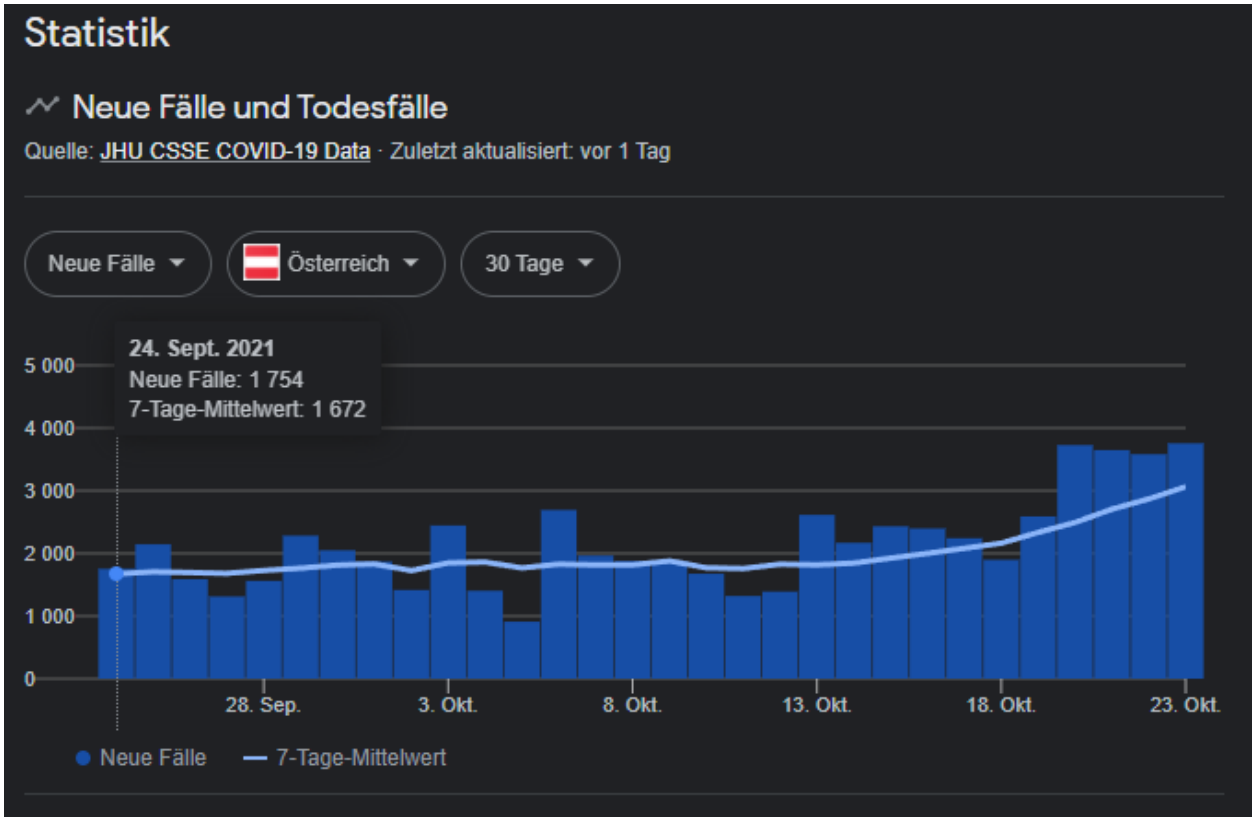


Abbildung 2-10: Neue Corona Fälle (Quelle: Google)

Abgeleitet von Abbildung 2-10 sieht man, dass ein berechtigter Ausschlag am 20.Oktober erkannt wurde, da die Zahl der neu Infizierten sprunghaft angestiegen und die 3000er Marke übertroffen wurde.

Da ein Reiz erkannt wurde, stellt sich die Frage, ab wann ist ein Reiz ein Reiz und muss reagiert werden?

Um die Frage „wann ein Reiz ein Reiz ist“ beantworten zu können, kann man einfache statistische Hilfsmittel verwenden. In der folgenden Abbildung werden Anhand des Median Schwellwerte berechnet und sobald ein Schwellwert überschritten wird, wird ein Reiz als Reiz erkannt.

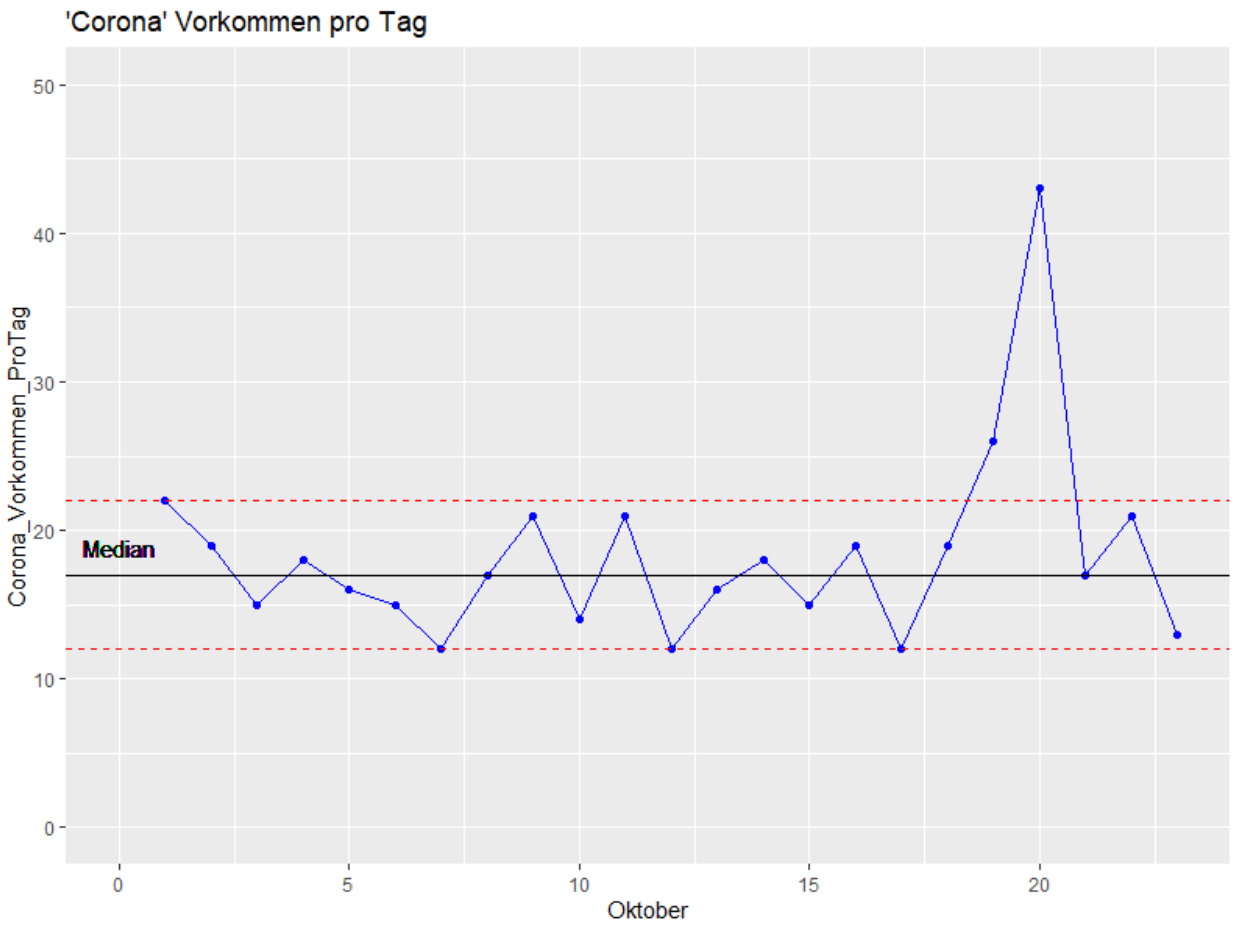


Abbildung 2-11: Schlüsselwort 'Corona' Vorkommen pro Tag mit Median und Schwellwerten (Quelle: eigene Abbildung)

Die zwei roten Linien zeigen den Schwellwert für diese Zeitreihe an. Bis jetzt wurde immer vom 20.Oktober gesprochen, dass dieser Tag so außergewöhnlich ist, jedoch wurde der Schwellwert am 19.Oktober schon überschritten. Somit wurde ein Reiz schon am 19.Oktober erkannt.

Die Frage was mit diesem Reiz geschieht, beziehungsweise wie er verarbeitet werden soll, wird in einem späteren Kapitel beschrieben, jedoch sei schon mal so viel gesagt, dass die Schlüsselwörter immer spezifisch auf Projekte angepasst werden müssen, da jedes Projekt unterschiedliche Risikofaktoren besitzt.

Die Zeitreihenanalyse kann für das Vorgehensmodell in gewisser Weise als Reizerkennung dienen, wenn die Schlüsselwörter und Schwellwerte genau genug spezifiziert werden. Wie kann ein Reiz nun als positiv oder negativ eingeschätzt werden? Bei manchen Reizen reicht die Überschreitung des unteren beziehungsweise des oberen Schwellenwertes aus um Aussagen treffen zu können, dass es ein positiver oder negativer Reiz ist. Bei manchen Überschreitungen ist das jedoch nicht der Fall und man kann sich eines weiteren Analyseverfahrens bedienen: Die Sentiment Analyse.

2.4.2 Sentiment Analyse

Sentiment Analyse oder Opinion Mining ist eine computergestützte Untersuchung von Texten. Sie ist eines der aktivsten Forschungsgebiete in den Bereichen natürliche Sprachverarbeitung, Data Mining, Informationsbeschaffung und Web-Mining (Cambria, Das, Bandyopadhyay, & Feraco, 2017). Positive Wörter für eine Sentiment Analyse sind zum Beispiel: beliebt, spitze, Unterstützung, einzigartig, etc. und negative Wörter sind zum Beispiel: absacken, Tod, rebellieren, zersetzen, etc..

Bleibt man bei dem oben genannten Beispiel mit dem Schlüsselwort „Corona“, so kann für alle Artikel, in denen das Schlüsselwort vorkommt, eine Sentiment Analyse gemacht werden. Es werden die Artikel „tokenisiert“ und mit einem Sentiment Lexikon verglichen. Für die deutsche Sprache gibt es in der Programmiersprache „R“ das Paket „pradadata“ mit der Funktion *germanlex*, welche über 8.800 Einträge besitzt. Das Format des Lexikons wird in folgender Abbildung gezeigt.

	word	qualifier	polarity_strength	pos
1	fehlschlagen	NEG	0.7	verben
2	rückläufig	NEG	0.7	adj
3	besiegen	POS	0.0	verben
4	vermindern	INT	0.5	verben
5	aufhören	SHI	0.0	verben

Abbildung 2-12: Format "germanlex" (Quelle: eigene Abbildung)

Das „germanlex“ Paket ist aufgeteilt in 4 Spalten:

- word
  zeigt Wörter an, auf welche in den Artikeln geachtet werden muss
- qualifier
  "NEG" für "negativ", "POS" für "positiv", "NEU" für "neutral", "SHI" für "shifter" (Ableitungen, die die Stimmung verstärken, zum Beispiel "schrecklich falsch", "wirklich gut") oder "INT" für "intensifier", wobei INT<1 ein Reduktionsfaktor und INT>1 ein Verstärkungsfaktor ist.
- polarity_strength
  Stärke der Polarität
- pos
  Wortart

Wenn man dieses Lexikon mit den, mit einem tokenisierten Datenframe verbindet und nach den „qualifier“ Parameter zählt, bekommt man folgenden Output:

```
Joining, by = "word"
  qualifier   n
1       POS 188
2       NEG 147
3       SHI 138
4       INT  58
5       NEU  49
```

Abbildung 2-13:Sentiment Beispiel (Quelle: eigene Abbildung)

In dieser Abbildung sieht man, dass die positiven Wörter in den Artikeln überwiegen, jedoch sind steigende Neuinfektionen kein positiver Grund. Was ist geschehen?

Das „germanlex“ Lexikon versucht so allgemein wie möglich zu sein und hat somit Wörter wie „steigend“, „stetig“ etc. mit einem POS *qualifier* hinterlegt. Der Satz: „Die Corona Neuinfektionen sind stark steigend“ wird somit als positives Sentiment angesehen, obwohl dies nicht der Fall ist. Aus diesem Grund müssen nicht nur die Schlüsselwörter, sondern auch die Lexika für die Sentiment Analysen projektspezifisch angepasst werden.

2.4.3 Kookkurrenz

In der allgemeinen Linguistik bezeichnet Kookkurrenz das gleichzeitige Auftreten zweier lexikalischer Einheiten in einer übergeordneten Einheit, beispielsweise in einem Satz oder einem Dokument. Treten diese beiden Begriffe auffällig häufig zusammen auf, wird davon ausgegangen, dass die beiden Begriffe voneinander abhängig sind. (Zong, Xia, & Zhang, 2021)

Diese Kookkurrenz kann für das Vorgehensmodell ein wichtiger Faktor sein, um wichtige Informationen bezüglich verschiedener Schlüsselwörter zu bekommen. Die Kookkurrenz funktioniert wieder auf Basis der Tokenisierung, jedoch wird der Text nicht auf einzelne Wörter aufgeteilt, sondern auf Zwei. Wie bereits in Kapitel 2.3 beschrieben, kann die Tokengröße variieren. Die Kookkurrenz würde auch mit einer größeren Tokengröße als zwei funktionieren, jedoch wird in sämtlichen Literaturen nur von einer Tokengröße von zwei bis maximal drei gesprochen.

Die folgende Abbildung zeigt die Kookkurrenz zwischen Wörtern von Artikeln, welche mit dem Schlüsselwort „Corona“ vorselektiert wurden.

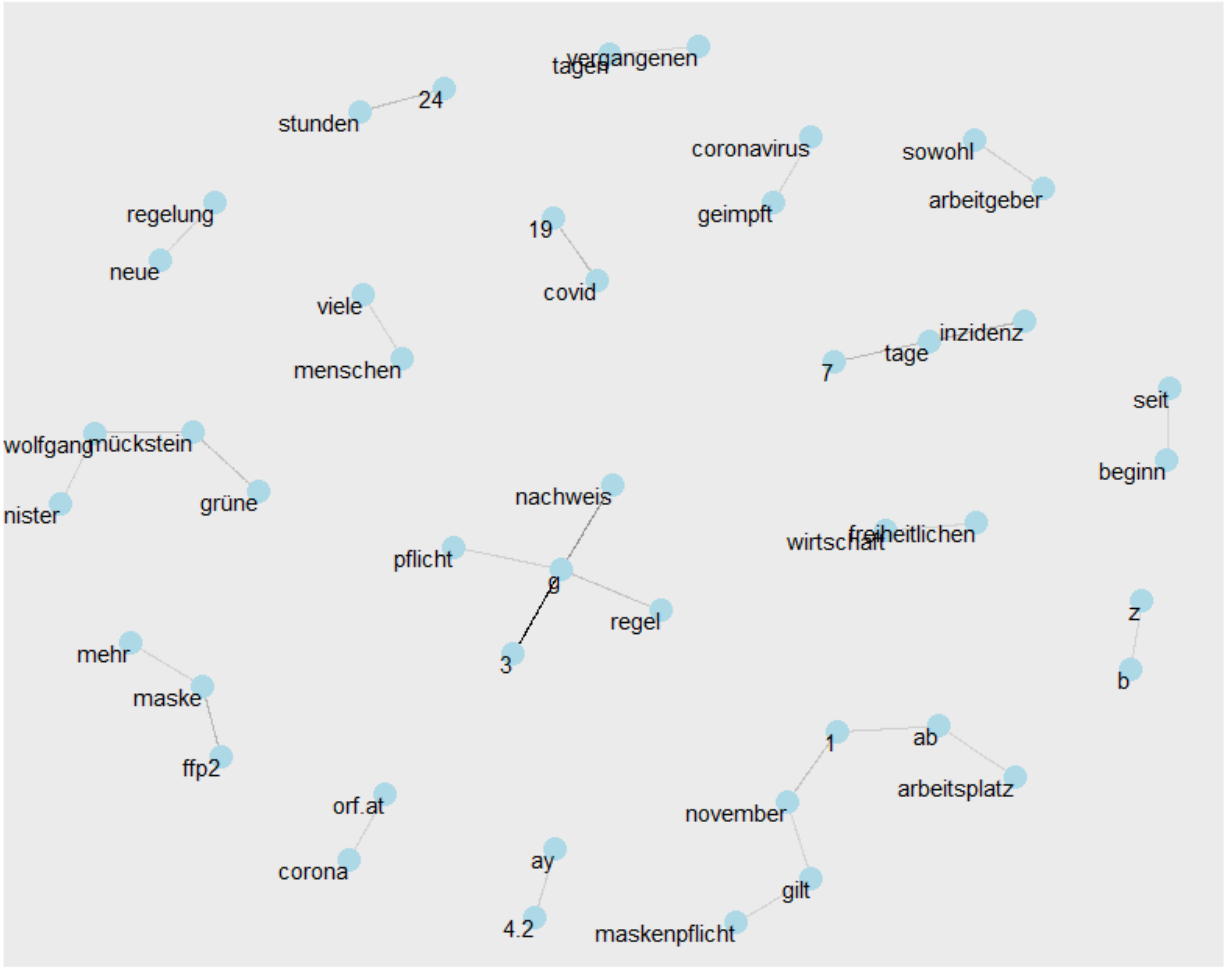


Abbildung 2-14: Kookkurrenz von Wörtern (Quelle: eigene Abbildung)

In dieser Abbildung sieht man sehr gut Abhängigkeiten zwischen verschiedenen Wörtern. Am auffälligsten, durch den dunklen schwarzen Strich, ist die Verbindung zwischen „3“ und „g“.

Ein anderes Beispiel für die Erkennung von Kookkurrenzen könnte folgendes sein: Man sucht nach dem Schlüsselwort „Förderung“ und bekommt als Wortpaarungen: Solaranlage – Strom, erneuerbare – Energie, etc. und wenn sich diese Wortpaarungen mit anderen Schlüsselwörtern eines projektspezifischen Schlüsselwortlexikons decken, könnte diese Erkenntnis relevant für das Vorgehensmodell werden.

Korrelationen

Die Tokenisierung nach Bigrammen ist eine nützliche Methode, um Paare benachbarter Wörter zu untersuchen. Jedoch kann man auch an Wörtern interessiert sein, die in bestimmten Dokumenten oder Kapiteln gemeinsam vorkommen, auch wenn sie nicht nebeneinanderstehen.

„Tidy“-Daten haben eine nützliche Struktur für den Vergleich zwischen Variablen oder Gruppierungen nach Zeilen, aber es kann eine Herausforderung sein, zwischen Zeilen zu vergleichen: zum Beispiel, um zu zählen, wie oft zwei Wörter im selben Dokument vorkommen, oder um zu sehen, wie sie korreliert sind. Bei den meisten Operationen zur Ermittlung von

Korrelationen müssen die Daten zunächst in eine breite Matrix umgewandelt werden. (Silge & Robinson, 2017)

2.4.4 Maschinelles Lernen

Maschinelles Lernen in Systemen bedeutet, dass Muster und Zusammenhänge in Daten erkannt werden, ohne explizit darauf programmiert zu sein.

Es gibt zwei Arten von diesem Verfahren, einerseits gibt es die Klassifikation und andererseits das Clustering.

- Clustering:
  Ziel der Clusteranalyse ist es, sinnvolle Gruppen in den Daten zu finden. Typischerweise werden in den Daten diese Gruppen in sich geschlossen und voneinander getrennt sein. Das Ziel ist es, Gruppen zu finden, deren Mitglieder etwas gemeinsam haben, was sie nicht mit den Mitgliedern anderer Gruppen teilen. Man bezeichnet diese Art „unsupervised Learning“. (Bouveyron, Celeux, Murphy, & Raftery, 2019)
- Klassifizierung ist ein Prozess der Kategorisierung eines gegebenen Datensatzes in Klassen, der sowohl bei strukturierten als auch bei unstrukturierten Daten durchgeführt werden kann. Der Prozess beginnt mit der Vorhersage der Klasse der gegebenen Datenpunkte. Die Klassen werden oft als Ziel, Label oder Kategorien bezeichnet. Bei der Klassifizierungsvorhersagemodellierung geht es um die Annäherung der Abbildungsfunktion von Eingangsvariablen auf diskrete Ausgangsvariablen. Das Hauptziel besteht darin, zu ermitteln, in welche Klasse/Kategorie die neuen Daten fallen werden. (Bouveyron, Celeux, Murphy, & Raftery, 2019)

Diese Analysen können für Vorhersagemodelle angewandt werden.

2.4.5 Lexika / Themenbereiche

Nun hat man verschiedenste Analyseverfahren, woher jedoch weiß man, welche Analyse für dieses Unternehmen beziehungsweise Projekt relevant ist?

Für jeden Themenbereich, welcher auf eine Anomalie getestet werden soll, müssen Lexika definiert werden. Falls ein Themenbereich zu umfangreich ist, kann es weiter aufgeteilt werden und es können weitere, spezifischere Lexika formuliert werden. Je spezifischer die Lexika definiert werden, desto genauer können Anomalien für ein Projekt oder einen Themenbereich erkannt werden.

Für Unternehmen können so Lexika für jede Hierarchiestufe eines Unternehmens definiert werden da jede Hierarchiestufe und jede Abteilung von anderen Einflussfaktoren abhängig ist.

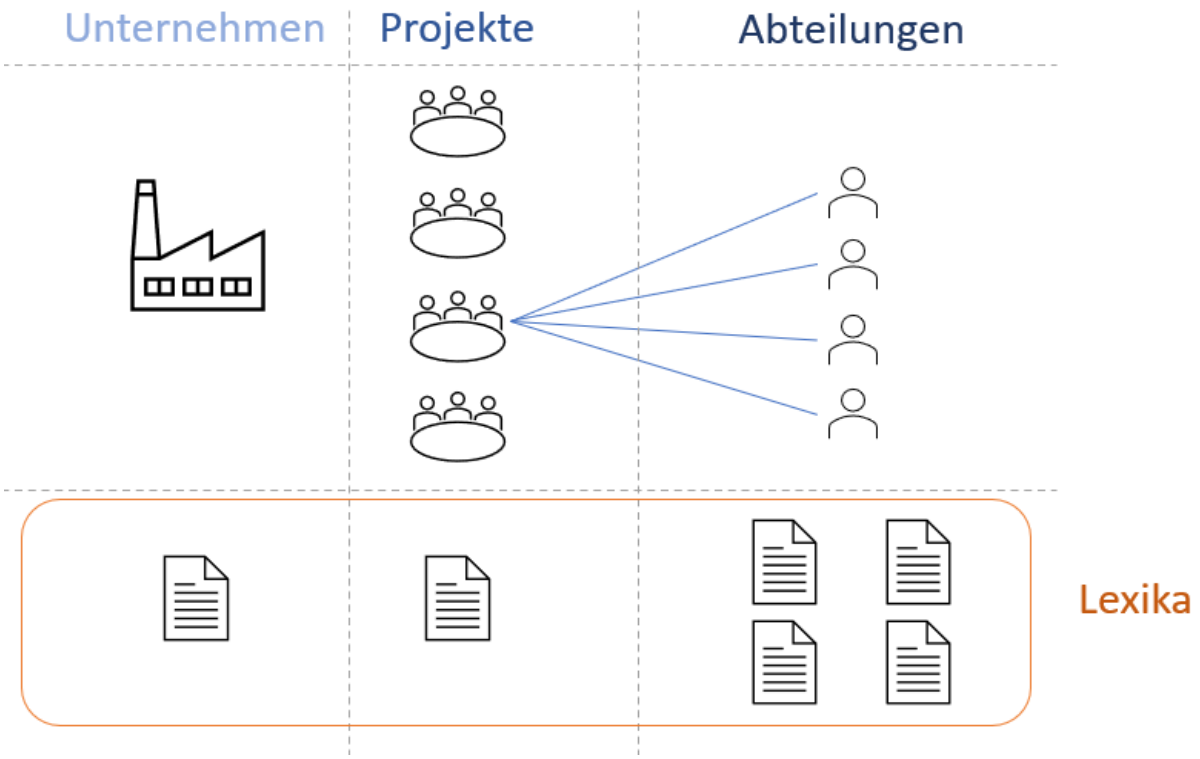


Abbildung 2-15: Mögliche Themenbereiche in einem Unternehmen (Quelle: eigene Abbildung)

3 PRAXIS

In diesem Kapitel wird das Wissen und die gewonnenen Erkenntnisse des theoretischen Rahmens verwendet. Es werden Wege der Datenbeschaffung aus verschiedenen Quellen erklärt und wie diese Daten in weiterer Folge analysiert werden können. Im Zuge des Vorgehensmodells wird ein Prototyp entwickelt, welcher für eine grafische Darstellung und automatische Auswertung der Daten sorgt.

Was muss das Vorgehensmodell beinhalten, dass die Forschungsfrage beantwortet werden kann?

3.1 Anforderungen

Im Folgenden werden die Anforderung an das Vorgehensmodell und den Prototyp definiert. Die Anforderungen wurden durch einen Brainstorming Prozess und dem Wissen des Theoretischen Rahmen erstellt.

1.1	**Daten einlesen**	Im Prototyp soll es möglich sein Daten des Typs „**csv**“ einzulesen.
1.2	Anomalie erkennen	Der Prototyp soll erkennen, wenn es eine Anomalie in den neuesten Daten gibt. Es soll auch ein Popup Fenster angezeigt werden, falls eine Anomalie erkannt wurde.
1.3	Lexikon (Schlüsselwörter) auswählen und anzeigen	Es muss er Benutzerin / dem Benutzer ermöglicht werden, zwischen verschiedenen Lexika, welche Schlüsselwörter beinhalten, auszuwählen und diese anzuzeigen.
1.4	Daten anzeigen	Es soll möglich sein, zumindest die ersten Datenreihen, der eingelesenen csv Datei anzuzeigen. Diese Daten sollen auch durch die Schlüsselwörter des Lexikons gefiltert sein.
1.5	Weltkarte anzeigen	Der Prototyp soll im Stande sein, die Anomalie auf einer Weltkarte mittels Heatmap anzuzeigen.
1.6	Datum auswählen	Es soll der Benutzerin / dem Benutzer möglich sein, ein Datum aus den eingelesenen Daten

		auszuwählen, woraufhin die Weltkarte auf diesen Tag aktualisiert wird.
1.7	Anzeigen der am meisten genannten Regionen	Es soll eine Tabelle angezeigt werden, mit den meistgenannten Regionen in Verbindung mit den Schlüsselwörtern des Lexikons
1.8	Zeitreihe	Es soll ein Diagramm dargestellt werden, welches die Summe der Erwähnungen pro Tag ausgibt (Erwähnungen sind vorkommen der Schlüsselwörter). Weiters soll der neueste Punkt in der Zeitreihe farblich markiert werden, wenn er außerhalb definierter Grenzen liegt.

Tabelle 3-1: Anforderungen an das Vorgehensmodell und den Prototyp

Da es sich im ersten Schritt um einen funktionalen Prototyp handelt, wird weniger Priorität auf Portabilität, Leistungseffizienz und Kompatibilität des Systems gelegt.

3.2 Programmiersprache R

Das Vorgehensmodell und der daraus resultierende Prototyp basieren auf der Programmiersprache „R“, da diese Programmiersprache sehr viele Werkzeuge für Analysen und Datenbeschaffung hat.

RStudio ist in Open-Source- und kommerziellen Editionen erhältlich und läuft auf dem Desktop (Windows, Mac und Linux) oder in einem Browser, der mit RStudio-Server oder RStudio-Workbench verbunden ist.

Der wichtigste Baustein für das Programmieren mit R sind die R-Pakete. Sie enthalten wiederverwendbare R-Funktionen, die Dokumentation, die beschreibt, wie sie zu verwenden sind, und Beispieldaten. (Wickham, R Packages, 2015)

Die verwendeten Pakete und deren Funktionen, werden im Laufe dieses Vorgehensmodells beschrieben.

3.2.1 Benutzeroberfläche

Die Programmiersprache R wird vorwiegend für Datenanalysen und Auswertungen verwendet und für die optische Darstellung kommt ein weiteres R-Paket zur Verwendung „shiny“.

Shiny ist ein R-Paket, mit dem sich interaktive Webanwendungen direkt aus R (RStudio) erstellen lassen. Es wurde in erster Linie für Datenwissenschaftler entwickelt, und zu diesem

Zweck ist es möglich browserbasierte Shiny-Anwendungen ohne HTML-, CSS- oder JavaScript-Kenntnisse zu erstellen. Es ist möglich Anwendungen, ohne diese Kenntnisse zu erstellen, jedoch ist es in Shiny möglich, diverse HTML-, CSS- oder JavaScript-Kenntnisse einzupflegen. (Wickham, Mastering Shiny, 2021)

3.3 Datenbeschaffung

Wie in dem Theorie Kapitel 2.2 über Datenmaterial bereits erwähnt, können Daten aus den verschiedensten Quellen beschaffen werden. In den folgenden Kapiteln werden Varianten beschrieben und aufgezeigt, wie eine mögliche Datenbeschaffung aussieht. Weiters wird erklärt, ob diese Daten zur weiteren Verwendung geeignet sind.

3.3.1 Daten von Online-Nachrichten

Diese Kapitel beschäftigt sich mit der Beschaffung von Informationen von Nachrichten Plattformen wie „orf.at“. Der wichtigste Bestandteil dieser Beschaffung ist der Zugriff auf eine API namens „NewsAPI“ (newsapi.org). Dieser Zugriff ist jedoch durch ein Preismodell eingeschränkt. Die genauen Restriktionen, können auf „newsapi.org/pricing“ nachgelesen werden. Für diesen Prototypen wird der gratis „Developer“ Zugriff verwendet. Dieser Zugriff beschränkt nicht nur die täglichen Anfragen auf die API, sondern auch die Länge der einzelnen Newsartikel (200 Zeichen) (NewsAPI, kein Datum). Um trotzdem die gesamten Artikel zu bekommen, kommen verschiedene R-Pakete zum Einsatz. Als Hilfestellung werden für die Beschreibung einzelner R-Pakete Abbildungen und Code-Snippets aufgeführt.

R-Paket „newsanchor“

Das Paket stellt eine Verbindung zu https://newsapi.org/ her. Die NewsAPI ist eine http-REST-API zum Suchen und Abrufen von Artikeln aus dem gesamten Web. Diese API ermöglicht es, Schlagzeilen und Artikeln von über 30.000 Nachrichtenquellen und Blogs abzurufen.

Dieses R-Paket hat verschiedene Funktionen, um spezifische Daten von der NewsAPI zu bekommen. Folgende Auflistung soll einen Überblick der Filter-Methoden des Pakets aufzeigen.

Es ist möglich nach:

- Schlagzeilen oder ganzen Artikeln,
- Suchwörtern (plus exkludierte Wörter),
- Sprache,
- Quelle,
- Domaine (oder exkludierte Domaine),

- Tag

Zu filtern. (Buhl, 2019)

Da der Prototyp Anomalien aus ungefilterten Informationenquellen erkennen soll, wird die Suchwort Filterfunktion nicht verwendet. In dieser Arbeit wird durch die Restriktionen der „Developer“ Lizenz nur eine Quelle verwendet.

Mit folgendem Code kann eine HTTP-REST-API Anfrage gesendet werden.

```
library(newsanchor)

news_orf = get_everything(query="", domains = "orf.at", from = 20220222,
to = 20220222, page_size = 100, api_key = "*******")
```

Listing 3-1: API Anfrage

Mit der Funktion *get_everything()* und den in Klammer gesetzten Parametern, bekommt man bis zu 100 Artikeln (page_size, auch durch Developer Lizenz auf maximal 100 Artikel begrenzt) von der Domaine *orf.at* und dem angegebenen Tag retour gesendet. Bei der „Developer“ Lizenz ist es jedoch nicht möglich Artikel zu suchen, welche älter als einen Monat alt sind. Bei dem Parameter „api_key“ wird der erlangte API-Schlüssel von NewsAPI eingegeben (Schulze, 2019).

In Abbildung 3-1 wird die Antwort der API-Anfrage gezeigt.

Name	Type	Value
news_orf	list [2]	List of length 2
metadata	list [1 x 8] (S3: data.frame)	A data.frame with 1 row and 8 columns
results_df	list [100 x 9] (S3: data.frame)	A data.frame with 100 rows and 9 columns
author	character [100]	'ORF.at' 'ORF.at' 'ORF.at' 'ORF.at' 'ORF.at' 'ORF.at' ...
title	character [100]	'UNO-Chef: Russische Soldaten sind keine Friedenswächter' 'Schwarzer Jogger ersc ...
description	character [100]	NA NA 'Nach dem Vorgehen Russlands in der Ostukraine bangt man in Taiwan, dass s ...
url	character [100]	'https://orf.at/stories/3248495/' 'https://orf.at/stories/3248494/' 'https://orf ...
url_to_image	character [100]	'https://orf.at/mojo/1_4_1/storyserver//news/common/images/og-fallback-news.png' ...
published_at	double (S3: POSIXct, POSIXt)	2022-02-22 23:09:11 2022-02-22 22:48:05 2022-02-22 22:21:26 2022-02-22 22:21:25 ...
content	character [100]	'UNO-Generalsekretär Antonio Guterres hat Russland für die Eskalation im Ukraine ...
id	logical [100]	NA NA NA NA NA NA ...
name	character [100]	'Orf.at' 'Orf.at' 'Orf.at' 'Orf.at' 'Orf.at' 'Orf.at' ...

Abbildung 3-1: Antwort der API (Quelle: eigene Abbildung)

In dieser Antwort sieht man eine Liste mit mehreren untergeordneten Listen.

- Results_df
  - Author – Autor des Artikels
  - Title – Titel des Artikels
  - url – die URL des Artikels
  - content – der Inhalt des Artikels

- published_at – Erstellungszeit des Artikels
- etc.

Da die Einschränkung der 200 Zeichen den Inhalt für den Prototypen verfälschen kann, muss ein weiteres R-Paket zur Hilfe genommen werden, um über den gesamten Artikel verfügen zu können. Für dieses Paket ist die URL des Artikels ausschlaggebend.

R-Paket „rvest“

Dieses Paket wird zur Informationsgewinnung von Webseiten verwendet. In Kapitel 2.2 wird ein kurzer Einblick über die Funktionsweise dieses R-Paketes gegeben.

Durch die erhaltenen URLs des „newsanchor“ Pakets kann mit den Funktionen des „rvest“ Pakets der gesamte Inhalt der Artikel generiert werden. Diese Art der Informationsgewinnung von Webseiten ist eine Kategorie des Data-Scrappings.

In seiner allgemeinsten Form ist Data-Scrapping eine Technik, bei der ein Computerprogramm Daten aus der Ausgabe eines anderen Programms extrahiert. Data-Scraping kommt häufig beim Web-Scraping vor. Dieser Prozess verwendet eine App, um wertvolle Informationen von einer Website zu extrahieren. (Munzert, Rubba, Meißner, & Nyhuis, 2015)

```
library(rvest)
library(lubridate)
library(tidyverse)
for (orf_url in news_orf$results_df$url){
  tryCatch({
  content = read_html(orf_url)
  date = html_nodes(content, ".print-only")
  title = html_nodes(content, ".story-lead-headline")
  location = html_nodes(content, xpath = '//meta[@property="og:locale"]')
   %>% html_attr('content')
  title = html_text(title)
  content = html_nodes(content, "p")[1:length(html_nodes(content, "p"))-1]
  content = html_text(content)
  content = paste(unlist(content), collapse = '')
  date = html_text(date)
  date = gsub("[\r\n]", "", date)
  date = str_trim(date, side = "both")
  date = as_datetime(date, format = "%d.%m.%Y %H.%M")
  language = 'de'
  orf_df = rbind(orf_df, data.frame(title, content, date, language,
orf_url, location))
  }, error = function(e){})
}
```

Listing 3-2: rvest - Data Scrapping

Mit dem Wissen aus Kapitel 2.2 können somit die Webseiten, welche hinter den URLs stecken, aufgeteilt werden und die wichtigsten Informationen extrahiert werden.
Wie in diesem Codebeispiel zu erkennen ist, werden 2 weitere Pakete (libraries) angeführt. Aus diesem Grunde eine kurze Beschreibung, wofür sie in diesem Beispiel verwendet werden.

- „lubridate“ Paket
    - R-Befehle für Datumszeiten sind im Allgemeinen nicht intuitiv und ändern sich je nach Art des verwendeten Datumsobjekts. Außerdem müssen die Methoden, die mit Datumszeiten verwenden werden, robust gegenüber Zeitzonen, Schalttagen, Sommerzeit und anderen zeitbezogenen Eigenheiten sein, und R fehlt es in manchen Situationen an diesen Fähigkeiten. Das Paket „lubridate“ hilft beim Umgang und der Interpretation von Datumszeiten.
- „tidyverse“ Paket
    - Das „tidyverse“ Paket ist eine Sammlung von R-Paketen und unterstützt eine Vielzahl von statistischen Analysen. Es ist rein für die Datenwissenschaft entwickelt worden. Alle Pakete haben eine gemeinsame Designphilosophie, Grammatik und Datenstrukturen um den Umgang der Funktionen so ähnlich wie möglich zu machen. In dem Codebeispiel **Fehler! Verweisquelle konnte nicht g efunden werden.**, wird jedoch nur die *pipe*-Funktion (%>%) verwendet, um weitere Aktionen mit dem Objekt durchführen zu können.

Die rbind() Funktion fügt die einzelnen Dataframes zusammen um sie in der gewünschten Struktur, als csv-Datei speichern zu können.

Die gespeicherte csv-Datei beinhaltet nun unter anderem die wichtigsten Daten wie:

- Titel,
- Inhalt,
- Erstellungsdatum,
- Sprache,

welche für die Auswertungen im Prototyp relevant sind.

3.3.2 Reddit Daten

In diesem und dem folgenden Kapitel bezüglich der Twitter Daten, wird die Vorgehensweise zur Datenbeschaffung gezeigt und welche Probleme dabei aufgetreten sind und ob diese Daten, für den Prototypen relevant sein können.

Die Webseite *reddit.com* hat bezüglich der Informationsgewinnung gegenüber den Nachrichten Daten und den Twitter Daten einen großen Vorteil. Es wird kein API-Schlüssel und somit keine Lizenz benötigt. Um zu veranschaulichen, wie ein Reddit Beitrag aufgebaut ist, soll nachstehende Graphik helfen.

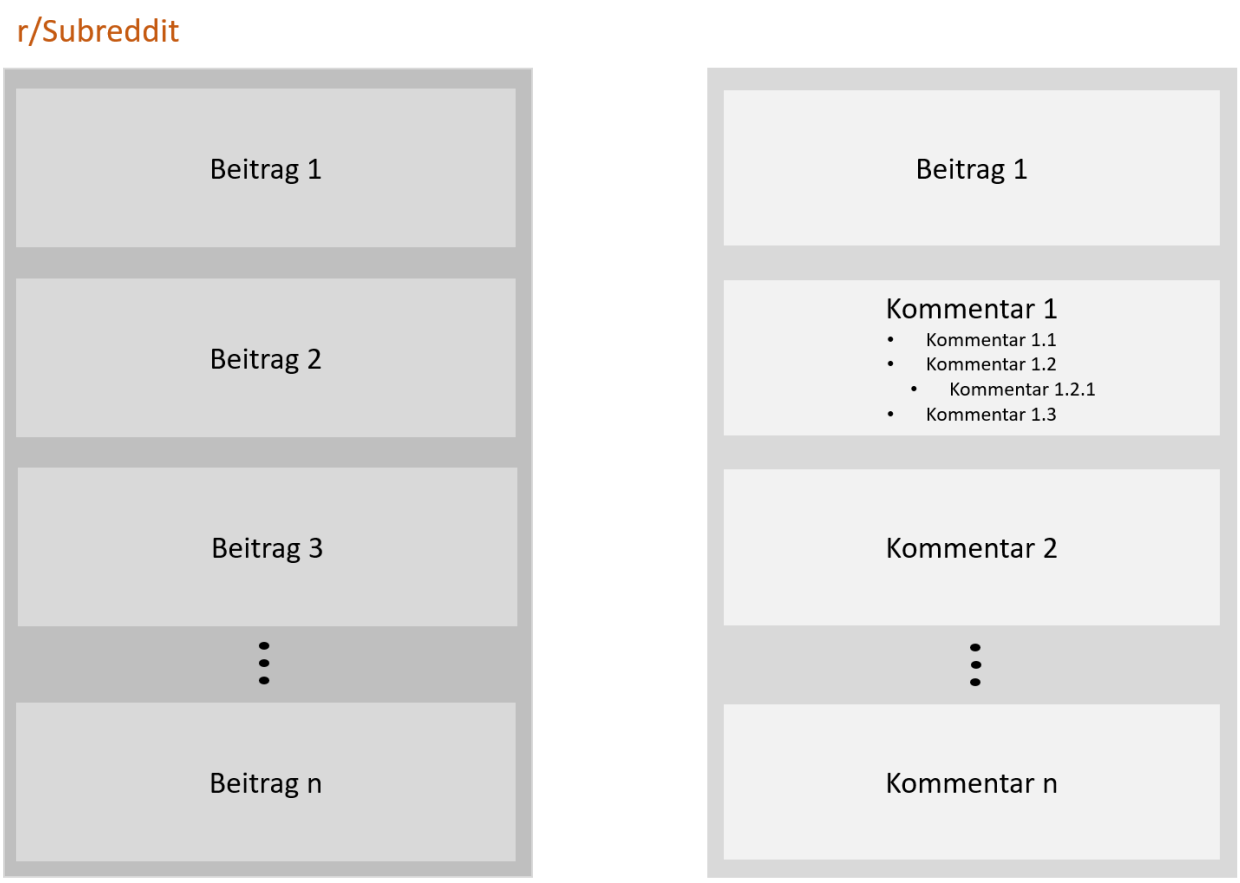


Abbildung 3-2: Reddit Beitrag Aufbau (Quelle: eigene Abbildung)

Die Plattform reddit.com ist aufgebaut durch verschiedene Subreddits. Subreddits sind Themenbereiche. In diesen Themenbereichen sollten alle Beiträge mit dem Thema des Themenbereichs zu tun haben. Zum Beispiel gibt es einen Subreddit namens CryptoCurrency. In diesem Themenbereich sollen sich alle Beiträge mit Kryptowährungen befassen.
Wie kann die Beschaffung der Beiträge eines solchen Subreddits nun aussehen?
Angenommen es sollen Daten aus dem Subreddit „CryptoCurrency“ extrahiert werden. Die URL für diesen Themenbereich lautet wie folgt: https://www.reddit.com/r/cryptocurrency/new/?limit=100&t=day.
Auf dieser Webseite werden die neuesten Beiträge in diesem Themenbereich angezeigt. Zum Extrahieren dieser Beiträge hat Reddit.com die Informationen dieser Seite auch in einem JSON[5]

[5] JavaScript Object Notation ist ein kompaktes Datenformat in einer einfach lesbaren Textform für den Austausch von Daten zwischen Anwendungen. JSON ist von Programmiersprachen unabhängig.

Format vorliegen, indem man die URL adaptiert. Mit den Parametern der Query[6] können die Limits der Beiträge und die Zeitspanne bestimmt werden. Die modifizierte URL könnte wie folgend aussehen: https://www.reddit.com/r/cryptocurrency/new.json?limit=100&t=day.
Da die Daten in einem JSON Format auf einer Webseite vorliegen, kommt ein weiteres R-Paket für die Extraktion zum Einsatz.

R-Paket „RJSONIO“

Dieses Paket, ermöglicht die Konvertierung in und aus Daten im Format JSON. Dadurch können R-Objekte in Javascript/ECMAScript/ActionScript-Code eingefügt werden und ermöglicht R-Programmierern das Lesen und Konvertieren von JSON-Inhalten in R-Objekte.

```
library(RJSONIO)
redditUrl =
https://www.reddit.com/r/cryptocurrency/new.json?limit=100&t=day

redditJSON = fromJSON(redditUrl)
```

Listing 3-3: RJSONIO Reddit Daten

Der vom JSON Format umgewandelte Output durch das „RJSONIO“ Paket sieht wie folgt aus:

Name	Type	Value
redditJSON	list [2]	List of length 2
kind	character [1]	'Listing'
data	list [6]	List of length 6
after	character [1]	't3_ti56a2'
dist	double [1]	100
modhash	character [1]	''
geo_filter	character [1]	''
children	list [100]	List of length 100
[[1]]	list [2]	List of length 2
[[2]]	list [2]	List of length 2
[[3]]	list [2]	List of length 2
[[4]]	list [2]	List of length 2

Abbildung 3-3: Output nach RJSONIO Umwandlung (Quelle: eigene Abbildung)

[6] Query wird der Teil eines URL-Pfades bezeichnet, mit dessen Hilfe man Informationen an und von Webseiten aus übermittelt.

Mit diesen R-Objekten kann die gewünschte Datenstruktur für den Prototypen erzielt werden. Wie dies bewerkstelligt werden kann, wird im folgenden Codebeispiel dargestellt.

```
library(lubridate)
for (redditRow in redditJSON$data$children) {
  title = redditRow$data$title
  content = redditRow$data$selftext
  date = redditRow$data$created_utc
  date = as_datetime(date)
  language = 'en'
  reddit_df = rbind(reddit_df, data.frame(title, content, date, language))
}
```

Listing 3-4: Reddit Dataframe

Somit liegt ein Dataframe in gewünschter Form vor, jedoch muss Abbildung 3-2 nochmal betrachtet werden. Jeder dieser Beiträge hat Kommentare und dessen Kommentare können weitere Kommentare haben. Laut eines Reddit Beitrages, wäre eine Kommentar Rekursion bis zu 10.000-mal möglich (Reddit, 2014).
Wie in Kapitel 2.2.2 erwähnt, können in social Media Netzwerken Einträge von Bots erstellt werden. Genau diese Propaganda der Bots tritt in den Reddit Daten sowie den Twitter Daten auf. Die meisten dieser Beiträge liefern keinen relevanten Inhalt. Weiters gibt es Einträge welche sarkastisch geschrieben sind und somit für das Vorgehensmodell nicht verwendbar sind. Selbes gilt auch für die Kommentare der Beiträge. Im Gegensatz dazu, liefern manche Kommentare wichtige Informationen, welche weiterverwendet werden könnten. Es werden im Laufe der Arbeit Versuche mit der Reddit URL getätigt, um brauchbare Daten zu generieren. Man kann folgende URL insofern ändern, dass nicht die neuesten Beiträge angezeigt werden, sondern die Beiträge, mit welchen am meisten interagiert (Anzahl an Kommentaren und Anzahl an Likes) wurden: https://www.reddit.com/r/cryptocurrency/top.json?limit=100&t=day. Mit dieser Herangehensweise filtert man zum einen die Beiträge der Bots heraus und zum anderen ist die Interaktion mit den Beiträgen ein Indikator für die Relevanz der Beiträge. Da die Top Beiträge von der täglichen Interaktion abhängig sind, können Beiträge auch über mehrere Tage in diesen Top 100 Beträgen verweilen. Weiters können diese Beiträge im Nachhinein editiert werden. Diese Faktoren können die Daten für den Prototypen verfälschen und eine Anomalie erkennen lassen, welche nicht vorhanden ist.

3.3.3 Twitter Daten

Wie bereits zu Beginn des vorherigen Kapitels erwähnt, werden Twitter Daten sowie die Online-Nachrichten Daten über eine API beschaffen.

Für den Zugriff auf die Twitter-API ist eine Registrierung für einen Twitter Entwickler Account nötig. Es gibt bei der Twitter-API mehrere Zugriffsebenen welche Restriktionen haben, wie viele Tweets (Text-Beiträge) man abrufen kann. Es gibt zwei Zugriffsebenen auf welche jeder Twitter Developer (registrierter Twitter Benutzer, der sich für einen Developer Account anmeldet) zugriff hat. Die zwei Zugriffsebenen heißen „essential“ und „elevated“. Der größte Unterschied zwischen diesen zwei Zugriffsebenen sind nicht die Anzahl der Anfragen, sondern die Rückgabe Objekte der Anfragen. Die Struktur und sogar die Klasse der Rückgabe Objekte sind unterschiedlich, worauf der ganze Code umgeschrieben werden muss, falls sich die Zugriffsebene ändert. Der „essential“ Account kann bis zu 500.000 Tweets pro Monat abrufen und der „elevated“ bis zu zwei Millionen. Der Zugriff auf zwei Millionen Tweets pro Monat klingt vorerst nach sehr viel, jedoch werden täglich über 500 Millionen Tweets gepostet (internet live stats, 2022).

In den folgenden Code-Beispielen werden die Suche und Extraktion von Twitter Daten dargestellt.

```
library(httr)
headers <- c(`Authorization` = sprintf('Bearer %s', bearer_token))
params = list(
  `query` = '#cryptocurrency OR #crypto lang:en',
  `max_results` = '100',
  `tweet.fields` =
'author_id,created_at,lang,geo,context_annotations,entities,public_metrics
,source',
  `start_time` = '2022-03-06T00:00:00Z',
  `end_time` = '2022-03-07T00:00:00Z'
)
response = httr::GET(url =
'https://api.twitter.com/2/tweets/search/recent',
httr::add_headers(.headers=headers), query = params)
```

Listing 3-5: Twitter Header und Parameter

Das „httr“- Paket wird verwendet um HTTP Anfragen zu senden wie GET(), POST(), etc. Im Fall der Datenbeschaffung wird nur der GET() Befehl verwendet um eine Anfrage an die Twitter-API zu senden. Um eine valide Antwort durch den GET() Befehl zu bekommen, muss ein HTTP-Header mit der entsprechenden Authentifizierung durch einen API-Schlüssel (bearer_token) mitgesendet werden.

Die „params“ Liste im Code-Beispiel gibt die verschiedenen Parameter an nach welchen Suchwörtern beziehungsweise Hashtags gesucht werden soll, wie viele Resultate pro Anfrage (100 ist das Maximum, jedoch können mit einem „next_token“ die nächsten 100 Resultate

retourniert werden) zurück geliefert werden sollen, die verschiedenen Felder (tweet.fields) welche retourniert werden sollen und der Zeitraum in dem die Tweets gepostet (Twitter, 2021).

Die Antwort (response) der Twitter Anfrage sieht wie folgt aus:

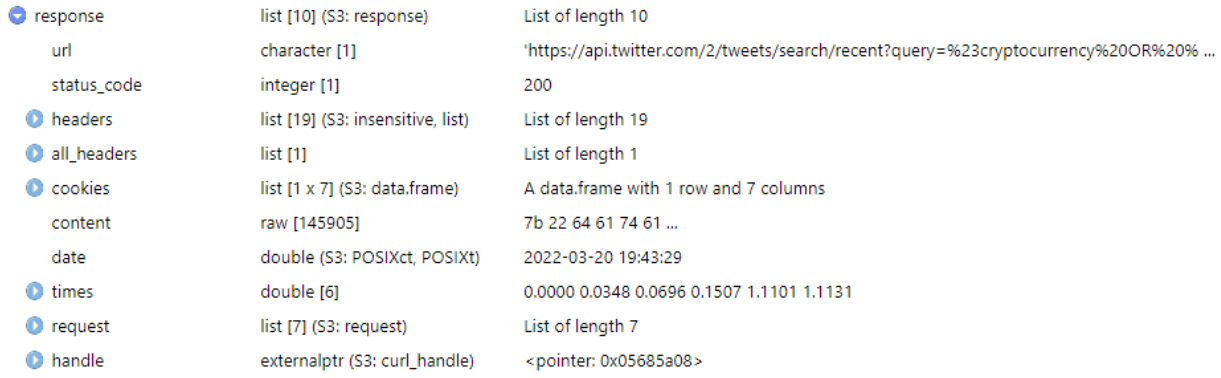


Abbildung 3-4: Antwort von Twitter API (Quelle: eigene Abbildung)

Da der gesamte Inhalt der Tweets im Hexadezimalsystem geliefert wird, kommt ein weiteres R-Paket zur Anwendung: „jsonlite“

```
library(jsonlite)
fas_body <-
  content(
    response,
    as = 'parsed',
    type = 'application/json',
  )
```

Listing 3-6: Umwandlung auf Dezimalsystem

Durch die Umwandlung des Hexadezimalsystems auf das Dezimalsystems kann auf die einzelnen Felder der Tweets zugegriffen werden, wie die folgende Abbildung und der darauffolgende Code darstellen:

Abbildung 3-5: Twitter API Response nach Umwandlung (Quelle: eigene Abbildung)

Mit for-Schleifen können wie in folgendem Code gezeigt, die wichtigsten Daten der Tweets extrahiert werden und in die passende Struktur für den Prototypen gebracht werden.

```
for (i in 1:length(fas_body$data)) {
    tweet_id = fas_body$data[[i]]$id
    auth_ID = fas_body$data[[i]]$author_id
    created_at = fas_body$data[[i]]$created_at
    text = fas_body$data[[i]]$text
    language = fas_body$data[[i]]$lang
}
```

Listing 3-7: Twitter Daten extrahieren

Mit den bisherigen Code-Beispielen der **Twitter Daten** bekommt man 100 Tweets zurück. Diese 100 Tweets sind in der Suchanfrage für „Crypto“ oder „Cryptocurrency“ gleichzusetzen mit 29 Sekunden. Soll bedeuten, dass in 29 Sekunden (von 23:59:30 bis 23:59:59 Uhr) 100 Tweets mit diesen Suchanfrage Wörtern getweetet worden sind. Um mehr Tweets zu bekommen, kommt der „next_token“ zum Einsatz, welcher bereits erwähnt worden ist. Dieser „next_token“ wird als weiteres Feld in den Parametern (params) hinzugefügt. Über eine for-Schleife kann so der „next_token“ immer neu eingesetzt werden. Mit dieser Möglichkeit wurden aus csv Daten mit 100 Tweets, csv Daten mit 40.000 Tweets. Diese 40.000 Tweets werden in durchschnittlich zweieinhalb Stunden gepostet. Nicht nur, dass die Aussagekraft von zweieinhalb Stunden Datenmaterial eher gering einzuschätzen ist, auch dauert die Extraktion dieser Daten mehr als 15 Minuten. Weiters sind die Daten mit Unicode versehen, welche Emojis darstellen sollen und sehr viele Symbole, welche vom Prototyp nicht interpretiert werden können. Durch den Einsatz der gsub() Funktion in R, kann durch reguläre Ausdrücke der Textinhalt der Tweets gefiltert und die Datenqualität verbessert werden. Der gewählte Output von 40.000 Tweets beruht auf zwei Tatsachen. Einerseits gibt es eine Rate von 450 Tweet-Suchanfragen pro 15 Minuten, welche in

einem Maximum von 45.000 Tweets resultiert und andererseits das Limit von 2 Millionen Suchanfragen pro Monat. 40.000 Anfragen mal circa 30 Tage im Monat ergibt 1.2 Millionen Anfragen. Mit dieser Anzahl bleiben noch genug Anfragen für weitere Entwicklungen dieser Datenbeschaffungsmethode übrig. Falls die Rate der Anfragen überschritten wird, können zwei mögliche Szenarien eintreten. Es werden nur etwa 10% der Angefragten Daten retourniert oder die Anfrage endet mit „Too Many Requests“ wie in folgender Abbildung dargestellt.

```
> fas_body
$title
[1] "Too Many Requests"

$detail
[1] "Too Many Requests"

$type
[1] "about:blank"

$status
[1] 429
```

Abbildung 3-6: Zu viele Anfragen (Quelle: eigene Abbildung)

Somit gibt es drei Varianten der Datenbeschaffung. Durch die Datenaufbereitung kann das Dataframe in ein gewünschtes Format gebracht werden, um vom Prototyp weiter verarbeitet und analysiert zu werden.

3.4 Datenverarbeitung und Analysen

In diesem Kapitel wird mit Hilfe eines Prototyps die Vorgehensweise zur Ergebnisgewinnung beschrieben. Es werden Code-Beispiele und Screenshots für eine visuelle Darstellung verwendet, um die Funktionsweise bestmöglich zu beschreiben. Da es sich um einen Prototyp handelt, hat die Funktionalität eine höhere Priorität als zum Beispiel die Wartbarkeit, Leistungseffizienz oder Kompatibilität.

3.4.1 Benutzeroberfläche „shiny“

In Kapitel 3.2.1 wird erklärt was „shiny“ ist und wofür es verwendet wird. Um die Funktionsweise von „shiny“ besser verstehen zu können, wird ein kurzer Überblick über die Struktur und die Grundlagen dieses R-Paketes gegeben.

Aufbau einer Shiny-App:

Shiny-Apps sind meist in einem einzigen Skript namens app.R enthalten. Das Skript app.R befindet sich in einem Verzeichnis (zum Beispiel appOrdner/) und die App kann mit runApp("appOrdner") ausgeführt werden.

app.R besteht aus drei Komponenten:

- ein Objekt der Benutzeroberfläche
- eine Server-Funktion
- einen Aufruf der Funktion shinyApp

Das Benutzeroberflächen (ui)-Objekt steuert das Layout und das Aussehen der Applikation. Die Serverfunktion enthält die Anweisungen, die der Computer benötigt, um die App zu erstellen. Schließlich erstellt die shinyApp-Funktion Shiny-App-Objekte aus einem expliziten UI/Server-Paar.

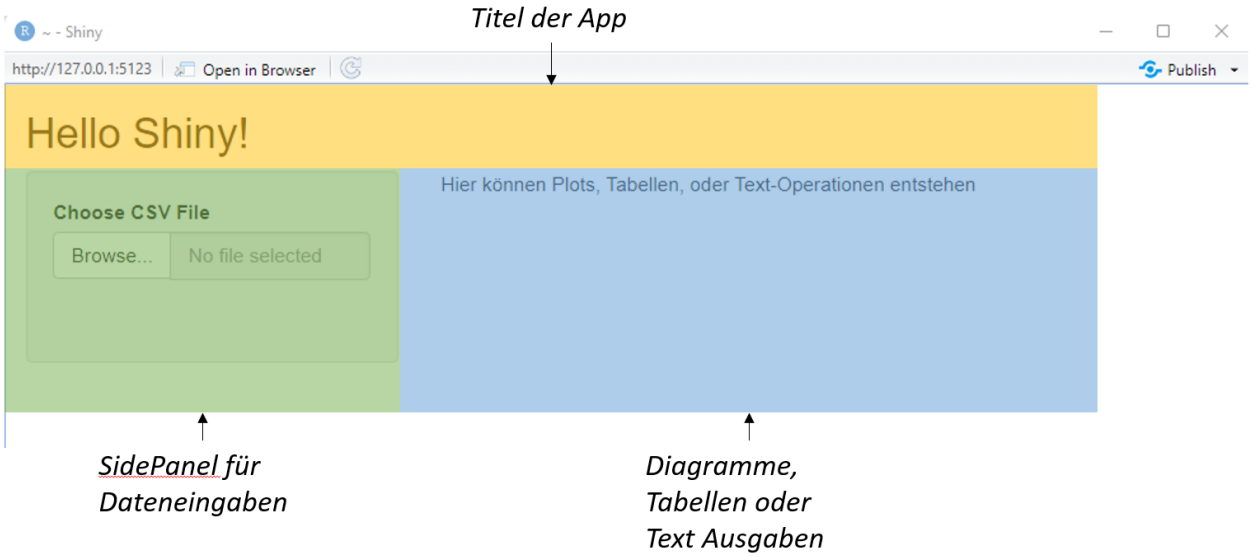


Abbildung 3-7: Aufbau der Benutzeroberfläche (Quelle: eigene Abbildung)

Der Aufbau der Benutzeroberfläche ergibt sich meistens aus drei Teilen. Dem Titel der Applikation, einem *sidebarPanel* welcher für diverse Dateneingaben verwendet wird und einem *mainPanel*, welcher für die Auswertungen zur Verfügung steht (Wickham, Mastering Shiny, 2021).

Der zugehörige Code für Abbildung 3-7 sieht wie folgt aus.

```
ui <- fluidPage(
  titlePanel("Hello Shiny!"),
  sidebarLayout(
    sidebarPanel(
      fileInput("file1", "Choose CSV File", accept = ".csv")
    ),
    mainPanel(
      textOutput("helloshiny")
    )
  )
)
server <- function(input, output) {
  output$helloshiny <- renderText({
    paste("Hier können Plots, Tabellen, oder Text-Operationen entstehen")
  })
}
shinyApp(ui = ui, server = server)
```

Listing 3-8: Grundlagen Benutzeroberfläche

Die Kommunikation zwischen der Benutzeroberfläche (ui) und dem Server (server) findet durch ID's statt. Die ID „helloshiny" im *mainPanel* wird in der Server Funktion mit „output$helloshiny" aufgerufen. Durch diese Kommunikation kann der Server verschiedene Operationen durchführen, um die Ausgabe zu verändern.

3.4.2 Daten Eingabe und sidebarPanel

Der *sidebarPanel* dient zur Eingabe der Daten. In diesem Abschnitt der Benutzeroberfläche können Benutzerinnen und Benutzer des Prototyps die Daten für die Auswertung auswählen. Der Datensatz muss eine Datei des Typs „csv" sein und die Struktur der Daten muss wie folgt aussehen.

	A	B	C	D	E	F
1	title	content	date	language	orf_url	location
2	Song-Contest-Sie	Die Band Maneskin a	5/23/2021 0:48	de	https://orf.a	de_DE
3	Leclerc muss um	Charles Leclerc hat si	5/22/2021 23:31	de	https://orf.a	de_DE
4	Tausende nach K		5/22/2021 23:20	de	https://orf.a	de_DE
5	„Ever Given"-Eig	Die Eigentümer des (	5/22/2021 23:19	de	https://orf.a	de_DE
6	Vulkanausbruch	In der Demokratisch	5/22/2021 22:21	de	https://orf.a	de_DE
7	Junger Kärntner	Das 13-jährige Kärntı	5/22/2021 21:44	de	https://orf.a	de_DE
8	Virgin Galactic ge	Das Raumflugzeug „V	5/22/2021 21:25	de	https://orf.a	de_DE
9	Budapester Stad	Der linksliberale Buc	5/22/2021 21:10	de	https://orf.a	de_DE
10	Polen kauft als e	Polen kauft als erste	5/22/2021 21:06	de	https://orf.a	de_DE
11	Atletico Madrid l	Atletico Madrid hat s	5/22/2021 20:05	de	https://orf.a	de_DE

Abbildung 3-8: Struktur der einzulesenden csv Datei (Quelle: eigene Abbildung)

In Abbildung 3-9 wird dargestellt, wie sich der *sidebarPanel* nach der Auswahl einer Datei ändert. Sobald man eine Datei ausgewählt hat, gibt es die Optionen sich die ersten paar Einträge der ausgewählten Daten anzeigen zu lassen, wenn die Schaltfläche „Dataframe anzeigen“ gedrückt wird. Da die Anomalie Erkennung auf verschiedene Themenbereiche spezifiziert werden kann, ist es essenziell den richtigen Themenbereich auszuwählen (Select Keyword Library). Auch dieser kann durch Klick auf die Schaltfläche angezeigt werden. In diesem Prototyp sind die Lexika der Themenbereiche fest codiert, was bedeutet, dass sie nur im Quellcode geändert werden können. Eine Option durch ein weiteres Hochladen einer csv Datei wird in diesem Prototyp noch nicht implementiert.

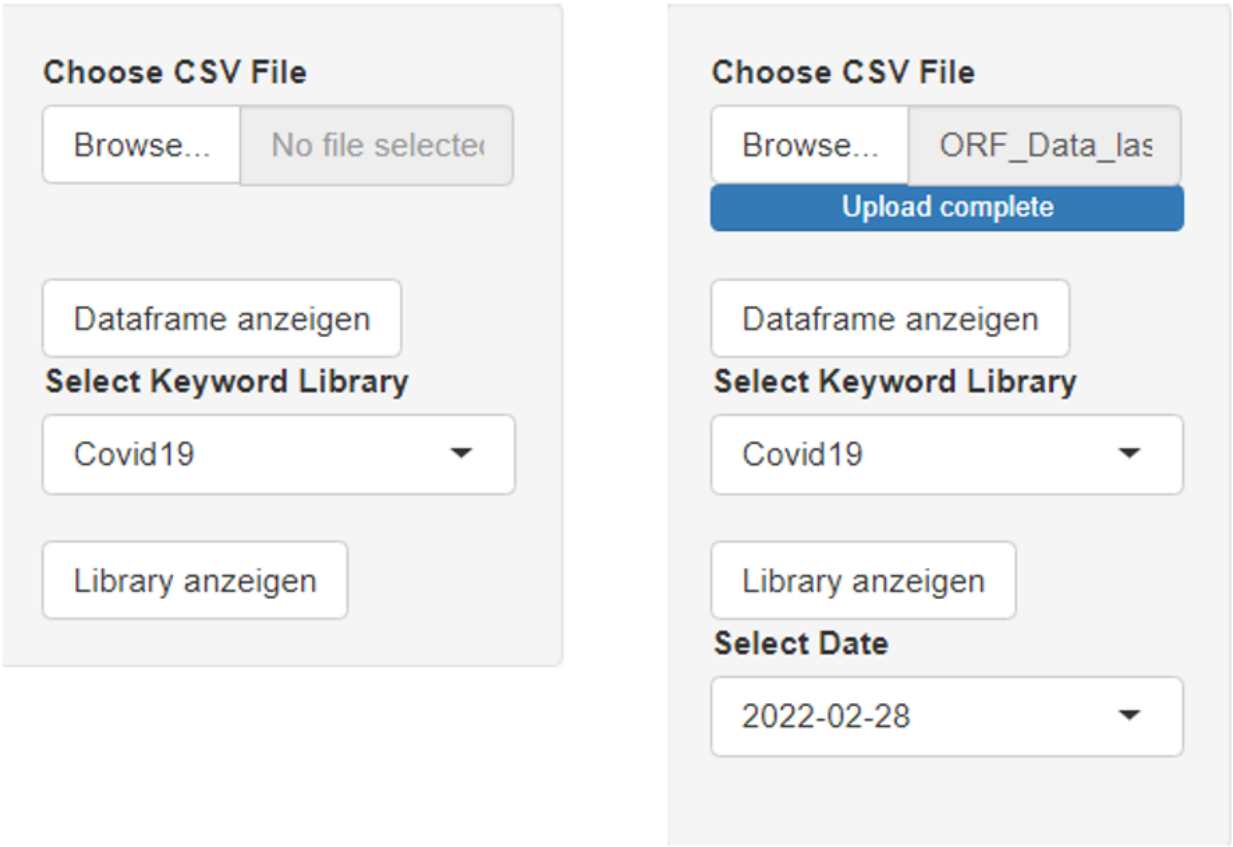


Abbildung 3-9:Prototyp sidebarPanel vor und nach Auswahl der Daten (Quelle: eigene Abbildung)

Wenn die Abbildung 3-9 betrachtet wird, sieht man, nach der Auswahl einer Datei, ein weiteres Feld zur Auswahl. Das Feld (Select Date) dient zur Auswahl des gewünschten Tages. Dieses Feld entsteht erst nach dem Einlesen der Daten da das Datum aus den Daten erst extrahiert werden muss. Durch die Auswahl des Tages können Anomalien retrospektiv erkannt werden. Automatisch wird immer der aktuellste Tag ausgewählt.

```
    sidebarPanel(
      fileInput("file1", "Choose CSV File", accept = ".csv"),
      actionButton("showcontent", "Dataframe anzeigen"),
      selectInput("library", "Select Keyword Library",
        list("Covid19", "Ressourcenknappheit", "Kryptowährungen")),
      actionButton("showLibrary", "Library anzeigen"),
      uiOutput("day")
    )
```

Listing 3-9: sidebarPanel Benutzeroberfläche

Die Funktion fileInput() öffnet ein Explorer Fenster um nach einer Datei zu suchen. Diese muss jedoch vom Typ *csv* sein. fileInput() speichert jedoch nur den Pfad zur Datei und lädt sie nicht direkt. Das Laden der Datei findet in der Serverfunktion statt. Die Funktionen hinter den actionButton() Schaltflächen und anderen Benutzerfläche-Objekten werden auch in der Serverfunktion ausgeführt.

CSV Datei einlesen. In dem folgenden Beispiel kommt ein weiterer wichtiger Baustein des „shiny“ Paketes vor. Die Serverfunktion reactive(). Reaktive Ausdrücke sind Ausdrücke, die reaktive Werte lesen und andere reaktive Ausdrücke aufrufen können. Wenn sich ein reaktiver Wert ändert, werden alle reaktiven Ausdrücke, die von ihm abhängen, als "ungültig" markiert und bei Bedarf automatisch neu ausgeführt. Wenn ein reaktiver Ausdruck als ungültig markiert ist, werden auch alle anderen reaktiven Ausdrücke, die ihn kürzlich aufgerufen haben, als ungültig markiert. Auf diese Weise wirken sich Ungültigkeitserklärungen auf alle Ausdrücke aus, die voneinander abhängen.
Somit muss die Datei jedes Mal neu geladen werden, wenn sich der Pfad von fileInput() in der Benutzeroberfläche ändert.

```
  csvdata = reactive({
    file <- input$file1
    ext <- tools::file_ext(file$datapath)
    req(file)
    validate(need(ext == "csv", "Please upload a csv file"))
    table_var = read.csv(file$datapath, header = TRUE)
  })
```

Listing 3-10: csv-Datei einlesen

In diesem Code wird nicht nur der Pfad für die Datei verwendet, um die Datei einzulesen, es wird auch auf die Endung des Pfades geachtet, ob es sich um den richtigen Datentyp handelt (validate()). Falls der richtige Datentyp validiert ist, wird die Datei mit read.csv() eingelesen. Da die Daten über eine Kopfzeile verfügen, wird das beim Einlesen angegeben (header = TRUE).

```
selectedLibrary = reactive({
  req(input$library)
  if (input$library == "Covid19") {
    keywordLibrary = covid19
  }
  if (input$library == "Kryptowährungen") {
    keywordLibrary = crypto
  }
  if (input$library == "Ressourcenknappheit") {
    keywordLibrary = ressources
  }
  keywordLibrary
})
```

Listing 3-11: Themenbereichswahl

Die Auswahl des Themenbereiches findet wiederum mit einer reactive() Funktion statt, um denselben Datentyp auf verschiedene Lexika zu prüfen. In dieser Serverfunktion wird auf die Auswahl von selectInput() geachtet. Hinter den Variablen covid19, crypto und ressources sind die fest codierten Lexika Daten in Form eines Dataframes gespeichert.

```
output$day = renderUI({
  req(csvdata())
  selectInput("v_date", "Select Date", choices =
    sort(unique(as.Date(csvdata()$date)), decreasing = TRUE))
})
```

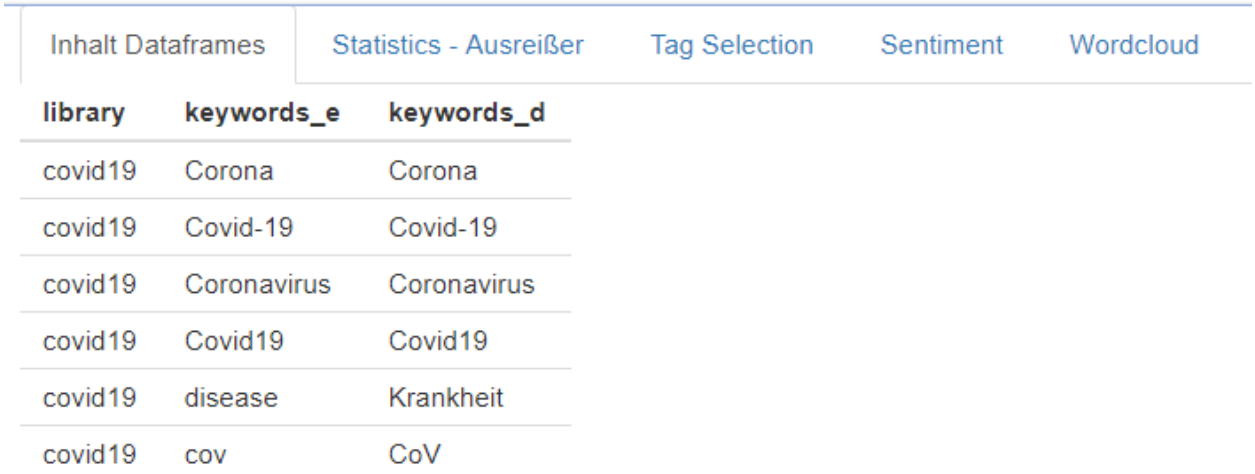
Listing 3-12: Serverfunktion für Auswahl des Tages

Da die Tage im Vorhinein nicht bekannt sind, muss die Benutzeroberfläche erst in der Serverfunktion erstellt werden. Diese zu erstellen benötigt den eingelesenen Datensatz (req(csvdata()). Danach wird mit selectInput() das Auswahlfeld mit den Datumsdaten der Datei generiert. Durch sort(x , decreasing = TRUE) wird automatisch der aktuellste Tag ausgewählt.

3.4.3 Themenbereich und Dataframe kontrollieren

Um verifizieren zu können, ob die richtige Struktur im Dataframe vorhanden ist, kann das Dataframe angezeigt werden. Weiters kann kontrolliert werden, nach welchen Schlüsselwörtern im Themenbereich gesucht und gefiltert wird.

Ein Beispiel für das hinterlegte Dataframe eines Themenbereiches wird in folgender Abbildung gezeigt:



library	keywords_e	keywords_d
covid19	Corona	Corona
covid19	Covid-19	Covid-19
covid19	Coronavirus	Coronavirus
covid19	Covid19	Covid19
covid19	disease	Krankheit
covid19	cov	CoV

Abbildung 3-10: Dataframe eines Themenbereiches (Quelle: eigene Abbildung)

In dieser Prototyp Version wird nur auf die deutschsprachigen Schlüsselwörter (keywords_d) Bezug genommen. In zukünftigen Versionen könnte eine automatische Spracherkennung des Textes implementiert werden, um die Textsprache identifizieren zu können und die richtigen Schlüsselwörter für die Anomalie Erkennung zu verwenden. Hilfreiche R-Pakete für die Spracherkennung sind „textcat“ und „fasttext“.

Da die Tabelle des Dataframes erst nach Klick auf die Schaltfläche angezeigt werden soll, kommt eine neue Funktion von „shiny“ zur Geltung.
Das reaktive Programmier-Framework von „shiny“ ist in erster Linie für reaktive Ausdrücke (reactive()) und Aktionen mit Beobachtern (observer()) konzipiert, die auf eine Änderung ihrer Eingaben reagieren. Das ist oft das, was in Shiny-Apps gewünscht wird, aber nicht immer: Manchmal möchte man auf eine bestimmte Aktion der Benutzerin und des Benutzers warten, wie zum Beispiel das Klicken auf eine Schaltfläche (actionButton()), bevor man einen Ausdruck berechnet oder eine Aktion ausführt. Ein reaktiver Wert oder Ausdruck, der verwendet wird, um andere Berechnungen auf diese Weise auszulösen, wird als Ereignis bezeichnet.
Dieses Ereignis kann mit observeEvent() überwacht werden.

```
observeEvent(input$showcontent, {
  output$contents <- renderTable({
    if(is.null(input$file1))
      return(NULL)
    head(csvdata(), 2)
  })
})
observeEvent(input$showLibrary, {
  output$libraryOutput = renderTable({
    selectedLibrary()
  })
})
```

Listing 3-13: Serverfunktion zum Anzeigen der Dataframes

Da in der Benutzeroberfläche etwas angezeigt werden soll müssen Texte, Tabellen und Grafiken aus Rohdaten erstellt werden. Diese werden durch Funktionen wie:

- renderTable() – für die Erstellung einer Tabelle
- renderText() – für die Erstellung eines Textes und
- renderPlot() – für die Erstellung einer Grafik

generiert. Diese Erstellungsmethoden müssen jedoch mit den Objekten der Benutzeroberfläche übereinstimmen. renderTable() – tableOutput(), renderText() – textOutput(), renderPlot() – plotOutput().
Die if()-Funktion in **Fehler! Verweisquelle konnte nicht gefunden werden.** ist als Prävention e iner Warnung implementiert, da eine Meldung angezeigt wird, solange kein Datensatz ausgewählt ist. Head(csvdata(), 2) gibt die ersten zwei Reihen des Datensatzes zurück und selectedLibrary() das fest codierte Dataframe des Themenbereichs.

3.4.4 Zeitreihen – Anomalie

Wird eine Anomalie in einem Datensatz erkannt oder nicht? Dieser Frage geht dieses Kapitel nach. Eine Anomalie kann einem Reiz oder einem Ausreißer gleichen. Um die Reize so gut wie möglich zu erkennen, werden zwei verschiedene Methoden verwendet.
Einerseits wird eine LOESS Regressionslinie mit definierten Grenzen gewählt und andererseits eine von Boxplots bekannte IQR (Interquartilsabstand) Methode.

Die IQR – Methode nimmt sich den Interquartilsabstand zu Hilfe. Der Interquartilsabstand wird aus dem dritten und dem ersten Quartil berechnet. Quantile sind Werte von Daten, die in vier gleiche Teile aufgeteilt werden. Mit Quantilen können die Streubreite und die Zentraltendenz eines Datensatzes schnell bestimmt werden.

Das erste Quartil beinhaltet die ersten 25% der Werte der Zahlenreihe. Der Median ist der Wert in der Mitte der Zahlenreihe und das dritte Quartil beinhaltet 75% der Werte. Diese Werte müssen für diese Bestimmung nach Größe geordnet sein.

Die genaue und effiziente Erkennung von Ausreißern in einem Datensatz ist immer eine Herausforderung, da es keine allgemeingültige Definition für Ausreißer gibt. Folglich muss die Person, die sich mit diesem Problem befasst, eine passende Definition finden und eine akzeptable Methode zu entwickeln, die den Anforderungen entspricht (Suri, Murty M, & Athithan).

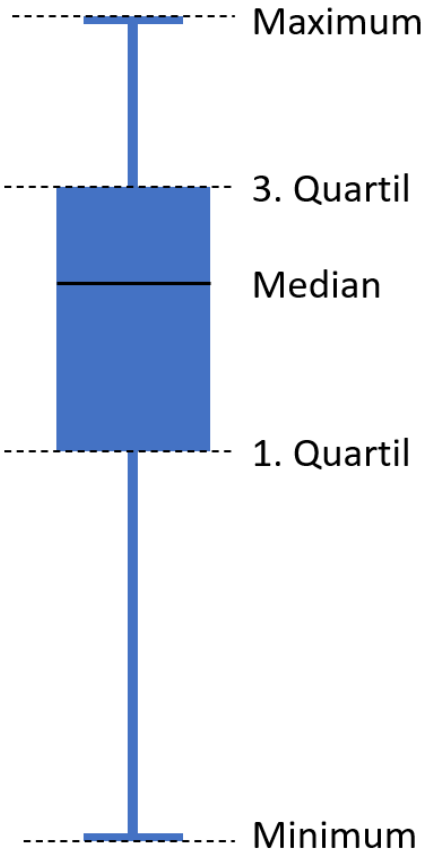


Abbildung 3-11: Quartile (Quelle: eigene Abbildung)

Wahl der Grenzen:

Untere Grenze: (Q1 - 1.5 * IQR)

Obere Grenze: (Q3 + 1.5 * IQR)

Diese Wahl der Grenzen beruht auf einem mathematischen Model, welches sich mit der Gaußschen Normalverteilung beschäftigt. Diese Formel hängt jedoch von der Verteilung der Daten ab. Wenn die Daten zum Beispiel einer Exponentialverteilung zu folgen scheinen, müsste die Formel angepasst werden (Chaudhary S. , 2019).

Eine Zeitreihe mit diesen Grenzen könnte wie folgt aussehen.

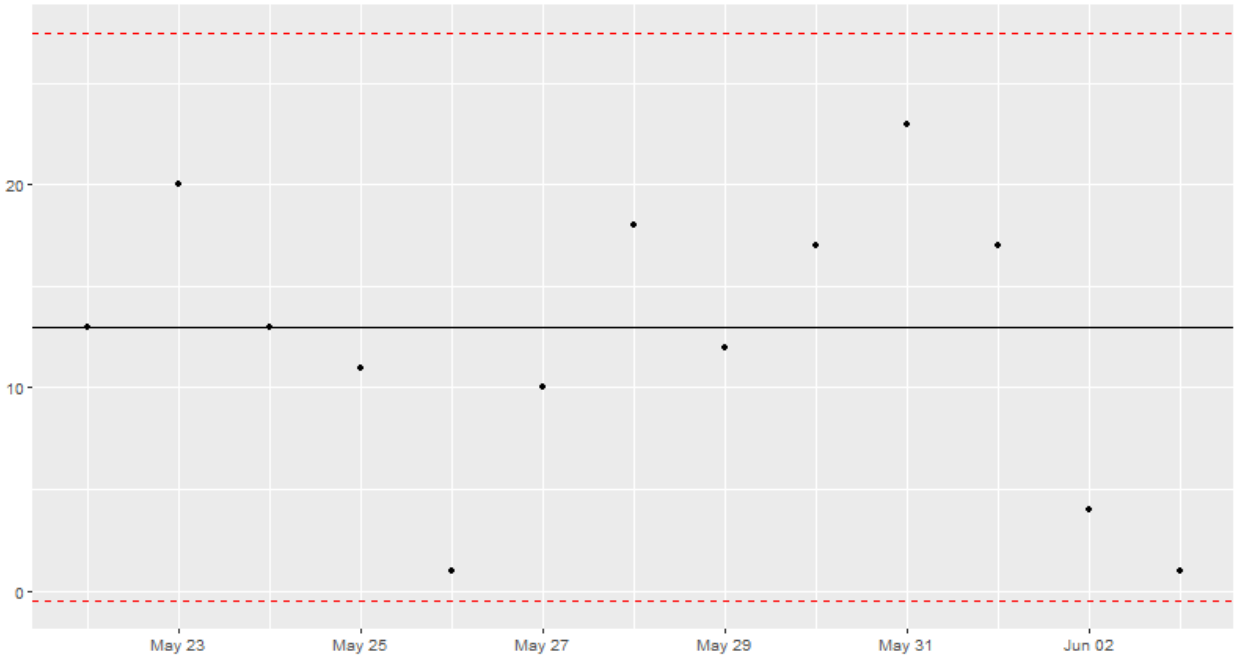


Abbildung 3-12: Zeitreihe mit IQR Methode (Quelle: eigene Abbildung)

Die zweite Methode ist eine Methode mit einer LOESS Linie:
LOESS ist eine von vielen "modernen" Modellierungsmethoden, die auf "klassischen" Methoden wie der linearen und nichtlinearen Regression oder der Methode der kleinsten Quadrate[7] aufbauen. Moderne Regressionsmethoden wurden entwickelt, um Situationen zu bewältigen, in denen die klassischen Verfahren nicht gut funktionieren oder nicht ohne unangemessenen Aufwand effektiv angewendet werden können. LOESS kombiniert die Einfachheit der linearen Regression der kleinsten Quadrate mit der Flexibilität der nichtlinearen Regression. Dies geschieht durch die Anpassung einfacher Modelle an lokalisierte Teilmengen der Daten, um eine Funktion zu erstellen, die den deterministischen Teil der Datenvariation Punkt für Punkt beschreibt. Einer der Hauptvorteile dieser Methode besteht darin, dass die Datenanalytikerin und der Datenanalytiker keine globale Funktion in irgendeiner Form angeben muss, um ein Modell an die Daten anzupassen (NIST/SEMATECH, 2013).
Kurz gesagt bekommt man mit der LOESS Funktion eine Kurve, welche sich an den Mittelwert der Daten von Teilsegmenten anpasst. Diese Funktionskurve kann mit einem Faktor multipliziert werden, um Grenzen zu definieren. Ein mögliches Diagramm mit Grenzen einer LOESS Funktion, könnte wie folgt dargestellt werden.

[7] Die Methode der kleinsten Quadrate, oder KQ-Methode ist das mathematische Standardverfahren zur Ausgleichungsrechnung. Dabei wird zu einer Menge von Datenpunkten eine Funktion bestimmt, die möglichst nahe an den Datenpunkten verläuft und somit die Daten bestmöglich zusammenfasst.

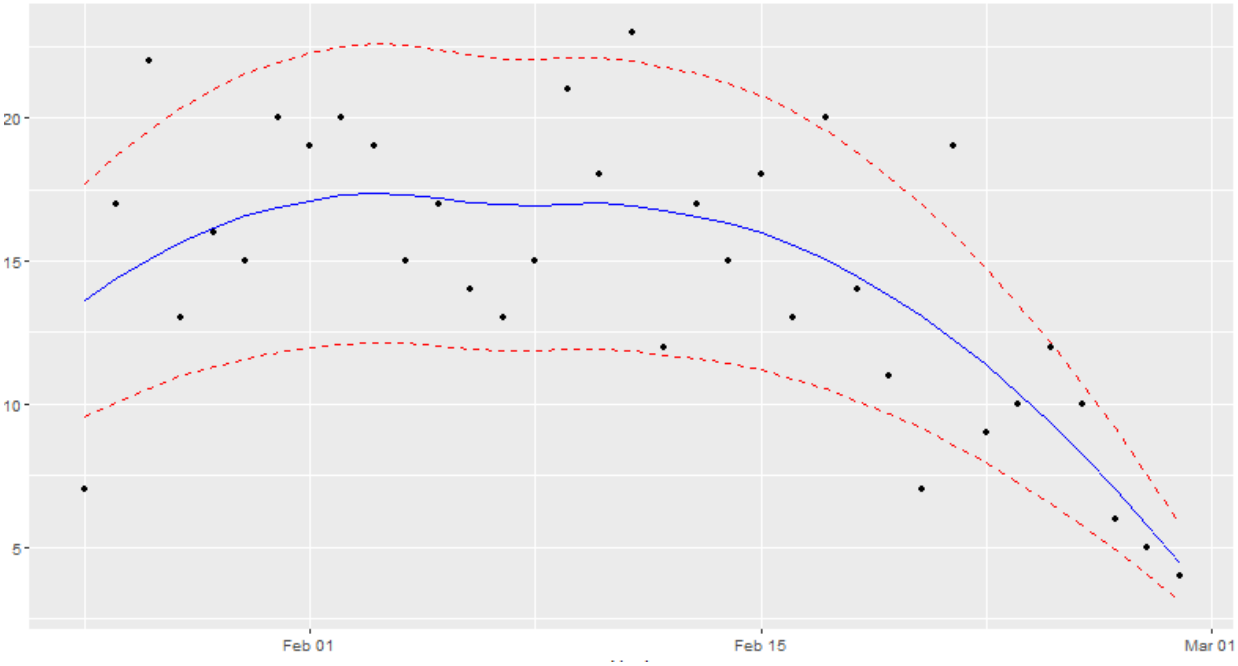


Abbildung 3-13: Zeitreihe mit LOESS Funktion (Quelle: eigene Abbildung)

Wie in dieser Abbildung ersichtlich, werden die Grenzen (rot strichliert) kleiner, wenn die Werte der Datenpunkte kleiner sind und es kann eine klare Tendenz festgestellt werden.

Für die Ausgabe der Diagramme wird ein weiteres R-Paket benötigt. Das Paket „ggplot2“. „ggplot2“ ist ein R-Paket für die Datenvisualisierung.
Jede ggplot2-Grafik hat drei Kernkomponenten:

- Ein Dataframe aus dem die Daten bezogen werden
- Ein Set von Verknüpfungen zwischen Variablen in den Daten und visuellen Eigenschaften (aesthetics)
- geom, ein geometrisches Layer-Objekt das beschreibt, wie jeder Datenpunkt gerendert (aufbereitet, wiedergegeben) wird.

Da für den das Diagramm ein nicht alle Daten relevant sind, wird der Datensatz nach den Schlüsselwörtern des ausgewählten Themenbereichs gefiltert. Diese Filterungsmethode wird in einer reaktiven Funktion abgebildet, da diese gefilterten Daten für alle weiteren Analysen benötigt werden.

```
filtered_data_library = reactive({
  req(input$library)
  req(csvdata())

  csvdata() %>% filter(grepl(paste(selectedLibrary()$keywords_d,
    collapse = " | "), content))
})
```

Listing 3-14: Daten nach Schlüsselwörtern filtern

Die Schlüsselwörter des ausgewählten Themenbereichs werden durch die paste() Funktion mit einem senkrechten Strich und zwei Leerzeichen vereint. Diese Reihenfolge erlaubt es, Vorkommen in den Inhalten der Daten zu suchen (grepl()). Die filter() Funktion retourniert nur die Daten, bei denen die Schlüsselwörter gefunden wurden.

Mit diesen gefilterten Daten kann die grafische Darstellung beginnen. Wie zuvor, wird zuerst das Diagramm mit der IQR-Methode beschrieben. In den ersten paar Zeilen des Code-Beispiels **Fehler! Verweisquelle konnte nicht gefunden werden.** werden die gefilterten Daten auf ein D ataframe reduziert, welches die Anzahl an Schlüsselwort Vorkommen pro Tag ausgibt. Angenommen man hat 100 Artikel oder Tweets pro Tag, so werden die Inhalte dieser Artikel nach Schlüsselwörtern des Themenbereichs durchsucht. Also gibt es zum Beispiel 13 Artikel, welche sich mit diesem Themenbereich Beschäftigen. Da der Inhalt dieser Artikel für die Zeitreihen irrelevant sind, werden neue Dataframes mit der Anzahl an Vorkommen gebildet. Dieser Schritt wird auch für die zweite Zeitreihenanalyse gemacht. Aus diesem Dataframe können dann die Quantile und der Median berechnet werden. Auch die Grenzen werden mit dem IQR Ausreißer Erkennungsmethode definiert. Die if() Funktion überprüft ob der Wert von dem ausgewählten Tag innerhalb der Grenzen liegt. Falls er außerhalb liegt, wird der Punkt auf der Zeitreihe rot markiert und eine Anomalie wurde erkannt.

```
output$timeline2 = renderPlot({
    req(filtered_data_library())
    req(input$v_date)
    temp = filtered_data_library()
    temp$date = as.Date(temp$date)
    temptable = table(temp$date)
    temptable = as.data.frame(temptable)
    temptable = temptable[as.Date(temptable$Var1) <=
      as.Date(input$v_date) ,]
    temptable$Var1 = as.Date(temptable$Var1)

    med = median(temptable$Freq)
    quantile1 = quantile(temptable$Freq, 0.25)
    quantile3 = quantile(temptable$Freq, 0.75)
    lower = quantile1 - 1.5 * IQR(temptable$Freq)
    upper = quantile3 + 1.5 * IQR(temptable$Freq)

    ggplot(temptable, aes(x = Var1, y = Freq))+
      geom_point() +
      {if(temptable[nrow(temptable), 2] > upper ||
        temptable[nrow(temptable), 2] < lower)
        geom_point(temptable, mapping = aes(x = Var1[nrow(temptable)],
        y = Freq[nrow(temptable)]), col = "red", size = 3)}+
      geom_hline(yintercept = med) +
      geom_hline(yintercept = lower, col = "red", linetype = "dashed") +
      geom_hline(yintercept = upper, col = "red", linetype = "dashed")
  })
```

Listing 3-15: Zeitreihe IQR-Methode

Der Code für die zweite Zeitreihe mit der LOESS Regressionslinie ähnelt dem Aufbau der IQR Methode sehr stark. Anstatt der Quantile und des Median, wird das Dataframe für die LOESS Regressionslinie verwendet. Die Faktoren, welche die Grenzen vorgeben sind in diesem Prototyp fest codiert, können aber für weitere Entwicklungen dynamisch gemacht werden, dass die Benutzerin und der Benutzer die Grenzen selbst definieren kann.

```
    loess_df = data.frame(Date = c(temptable$Var1), values =
      c(loess(formula = Freq ~ as.numeric(Var1),  data = temptable)[2]))
    upper = loess_df[nrow(loess_df),2]*1.3
    lower = loess_df[nrow(loess_df),2]*0.7

    ggplot(temptable, aes(x = Var1, y = Freq))+
      geom_point() +
      {if(temptable[nrow(temptable), 2] > upper ||
        temptable[nrow(temptable), 2] < lower)
        geom_point(temptable, mapping = aes(x = Var1[nrow(temptable)], y =
        Freq[nrow(temptable)]), col = "red", size = 3)}+
      geom_line(data = loess_df, aes(x = Date, y = fitted), col = "blue")+
      geom_line(data = loess_df, aes(x = Date, y = fitted*1.3), col =
        "red", linetype = "dashed") +
      geom_line(data = loess_df, aes(x = Date, y = fitted*0.7), col =
        "red", linetype = "dashed")
```

Listing 3-16: Zeitreihe LOESS-Methode

Zusätzlich zu den zwei Zeitreihen wird eine Tabelle mit den 5 aktuellsten Tagen und deren Anzahl an Vorkommen pro Tag ausgegeben.

```
output$vorkommen = renderTable({
  req(filtered_data_library())
  temp = filtered_data_library()
  temp$date = as.Date(temp$date)

  ordertemp = table(temp$date)
  ordertemp = as.data.frame(ordertemp)
  ordertemp = ordertemp[as.Date(ordertemp$Var1)
    <= as.Date(input$v_date) ,]
  names(ordertemp)[names(ordertemp) ==
    'Var1'] <- 'Datum'
  names(ordertemp)[names(ordertemp) ==
    'Freq'] <- 'Anzahl'
  head(ordertemp %>% arrange(desc(Var1)), 5)
})
```

Listing 3-17: Tabelle für Zeitreihenberechnung

Datum	Anzahl
2022-02-28	4
2022-02-27	5
2022-02-26	6
2022-02-25	10
2022-02-24	12

Abbildung 3-14:Vorkommen des Themenbereiches pro Tag (Quelle: eigene Abbildung)

3.4.5 Zentren in Daten erkennen - Weltkarte

In diesem Kapitel wird eine Analyse erstellt, um mögliche Zentren für den ausgewählten Themenbereich an dem ausgewählten Tag auszumachen.
Für ein besseres Verständnis was die Weltkarte darstellen soll, soll folgendes Beispiel helfen: Es gibt einen Datensatz von 100 Nachrichtenartikeln an dem ausgewählten Tag. Der Themenbereich beschäftigt sich mit Ressourcenknappheit und Lieferverzögerungen. So werden verschiedene Schlüsselwörter definiert, welche aussagekräftig für diesen Themenbereich sind. Der Datensatz wird nach diesen Schlüsselwörtern gefiltert. Von den 100 Nachrichtenartikeln behandeln 25 dieses Thema. Diese 25 Nachrichtenartikeln werden nun nach Ländernamen beziehungsweise deren Hauptstädten durchsucht. Diese Filtermethode hat den Hintergrund, dass oftmals der Name der Hauptstadt und nicht explizit das Land erwähnt wird in diversen Artikeln. Die Vorkommnisse werden dann in einer Wärmekarte auf einer Weltkarte aufgetragen. Gibt es zum Beispiel Lieferprobleme, weil ein Schiff im Suezkanal stecken bleibt, so wird öfter das Land Ägypten vorkommen und auf der Weltkarte hervorgehoben. So kann dementsprechend auf dieses Problem reagiert werden.

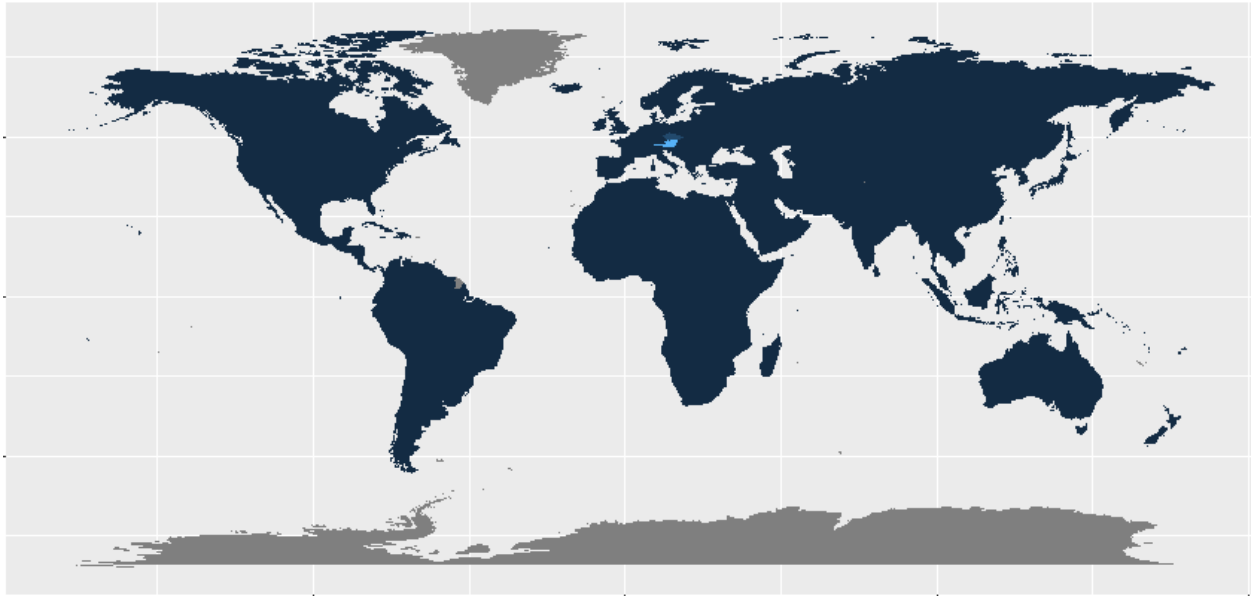


Abbildung 3-15: Weltkarte + Wärmebild (Quelle: eigene Abbildung)

In dieser Abbildung ist durch das Wärmebild zu sehen, dass es auf Basis des Themenbereichs ein häufigeres Vorkommen von Österreich beziehungsweise Wien in den Daten gibt.
Da die Daten durch die Tagesauswahl geändert werden können, kann zum Beispiel ein Verlauf oder die Ausbreitung einer Krankheit festgestellt und prognostiziert werden. Beispielsweise wird eine neue Mutation einer Krankheit festgestellt. Diese hat den Ursprung in England, geht über Belgien nach Deutschland und Frankreich. Dies kann im Laufe von paar Tagen geschehen und die Wärmebilder der einzelnen Tage geben Informationen für eine Interpretation der weiteren Ausbreitung.

Um diese Weltkarte inklusive Wärmebild zu generieren, werden drei verschiedene Datensätze benötigt und kombiniert. Der erste Datensatz liefert das R-Paket „ggplot2“. Dieses Paket beinhaltet ein eine Funktion *map_data(„world“)*. Diese Daten liegen in folgender Form vor:

	long	lat	group	order	region	subregion
1	-69.89912	12.45200	1	1	Aruba	NA
2	-69.89571	12.42300	1	2	Aruba	NA
3	-69.94219	12.43853	1	3	Aruba	NA
4	-70.00415	12.50049	1	4	Aruba	NA
5	-70.06612	12.54697	1	5	Aruba	NA
6	-70.05088	12.59707	1	6	Aruba	NA
7	-70.03511	12.61411	1	7	Aruba	NA

Abbildung 3-16: ggplot2 map_data() Dataframe (Quelle: eigene Abbildung)

In diesem Datenframe werden die Längengrade (long) und die Breitengrade (lat) angegeben, um ein Polygon zu generieren. Die Region (region) ist für das Erstellen des Polygons wichtig, die Längen und Breitengrade differenzieren zu können, zu welcher Region sie gehören.
Der zweite Datensatz ist die eingelesene *csv* Datei, welche bereits durch die Schlüsselwörter des Themenbereichs und der Tagesauswahl gefiltert ist.
Der dritte Datensatz ist wiederum eine *csv* Datei mit einer Lister der Staaten der Erde. Diese Datei beinhaltet den deutschen, englischen und lokalen Namen des Staates, die Hauptstadt (in Deutsch), Einwohnerzahl und Fläche. Für die weitere Verwendung dieses Datensatzes werden der deutsche und englische Name des Staates sowie die Hauptstadt verwendet. Der deutsche Name des Staates und die dazu gehörige Hauptstadt werden für das Wärmebild verwendet, da nach deren Vorkommen in den Artikeln gesucht wird. Der englische Name wird für die Verschmelzung mit dem ersten Datensatz (map_data()) verwendet.
Die Erstellung des Datensatzes, welcher danach mit dem map_data() Dataframe verschmelzt werden kann, sieht wie folgt aus.

```
df_for_worldmap = reactive({
    req(input$v_date)
    req(csvdata())
    req(csv_laenderliste)
    req(mapdata)

    df_laenderliste = NULL
    for(i in csv_laenderliste$Englischer.Name) {
      df_laenderliste[[i]] <- 0
    }
    df_laenderliste = as.data.frame(df_laenderliste)
    df.cbinded = cbind(filtered_data_library(), df_laenderliste)
    df.cbinded = df.cbinded[as.Date(df.cbinded$date) == input$v_date,]

    for (j in 1:length(df.cbinded$content)) {
      for (i in 7:ncol(df.cbinded)) {
        if(grepl(tolower(csv_laenderliste$Staat[i-6]),
        tolower(df.cbinded$content[j]))) {
          df.cbinded[[names(df.cbinded)[i]]][j] =
          df.cbinded[[names(df.cbinded)[i]]][j] + 1
        }
        else if(grepl(tolower(csv_laenderliste$Hauptstadt[i-6]),
        tolower(df.cbinded$content[j]))) {
          df.cbinded[[names(df.cbinded)[i]]][j] =
          df.cbinded[[names(df.cbinded)[i]]][j] + 1
        }
      }
    }
    df.land_counts = data.frame(region = csv_laenderliste$Englischer.Name,
      erwaehnungen = colSums(df.cbinded[,7:ncol(df.cbinded)]),
      row.names = 1:length(names(df.cbinded[,7:ncol(df.cbinded)])))
    df.land_counts
  })
```

Listing 3-18: Erstellung des Dataframes für die Weltkarte

Aus der *csv* Datei „Liste aller Staaten der Erde“ wird ein neues Dataframe erzeugt, welches für jedes Land eine eigene Spalte erstellt (Spaltenname = englischer Name des Staates). Diese Spalten werden mit 0 initialisiert, um später die Vorkommen in den Artikeln hinzufügen zu können.

Durch die cbind()-Funktion können unterschiedliche Dataframes kombiniert werden. Cbind() bedeutet, dass die Spalten eines Dataframes an das andere angeheftet werden. Sobald das gefilterte Dataframe mit dem Dataframe der Liste aller Staaten kombiniert ist, wird das kombinierte Dataframe in einer verschachtelten for-Schleife iteriert, um die Vorkommen der Länder oder der Hauptstädte zu zählen. Da die Daten nicht verfälscht werden, indem in einem

Artikel öfter das gleiche Land vorkommt, kann jedes Land pro Artikel nur einmal gezählt werden. Im Letzen Abschnitt des Codes wird aus dem iterierten Datenframe ein neues erstellt, indem alle Vorkommen zusammengezählt werden. Dieses Dataframe wird für die Darstellung des Wärmebildes verwendet.

```
output$selectedDateMap = renderPlot({
  req(df_for_worldmap())
  data = left_join(mapdata, df_for_worldmap(), by = "region")

  ggplot(data, aes(x = long, y = lat, group = group)) +
    geom_polygon(aes(fill = erwaehnungen))
})
```

Listing 3-19: Erstellen der Weltkarte mit Wärmebild

Das zuvor erarbeitete Dataframe wird mit einem left_join() mit den Daten des map_data(„world“) verschmolzen. Bei einem LEFT JOIN werden alle Datensätze der "Haupttabelle" (map_data()) mit denen des anderen Dataframes "links herum" verknüpft. Das bedeutet, dass alle Datensätze der "linken" Tabelle auf jeden Fall angezeigt werden.
Durch die ggplot2-Funktion geom_polygon() wird das Polygon für die Weltkarte erstellt und das Wärmebild durch die fill-Ästhetik (aes(fill =)).

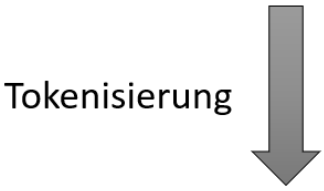
3.4.6 Sentiment bestimmen

Durch die bisherigen Schritte des Vorgehensmodells werden Daten nach Themenbereichen gefiltert, in einer Zeitreihenanalyse dargestellt, um Reize beziehungsweise Ausreißer zu erkennen und es werden diese Daten in Form eines Wärmebildes auf einer Weltkarte aufgetragen, um Zentren zu erkennen, welche Regionen von der Anomalie betroffen sind.

Da alle Daten aus Textdaten stammen, um eine Anomalie zu erkennen, kommt in diesem Kapitel ein weiterer Teil der Semantik vor. Die Sentiment Analyse. Wie im Theoretischen Rahmen beschrieben, beschäftigt sich die Sentiment Analyse mit der Stimmungserkennung. Somit kann automatisch bestimmt werden, ob eine Anomalie einen positiven oder negativen Einfluss auf diesen Themenbereich haben kann.

Die Ausgangsbasis für die Sentiment Analyse ist der nach Themenbereichen gefilterte Datensatz. Um das Sentiment bestimmen zu können, müssen alle Wörter in den Artikeln des Datensatzes mit einem vordefinierten Sentiment Lexikon abgeglichen beziehungsweise verschmolzen werden. Um die einzelnen Wörter mit den dem Lexikon vergleichen zu können, werden die Artikel „tokenisiert“, das heißt, die Artikel werden in einzelne Wörter aufgespalten. Wenn ein Artikel 150 Wörter hat, und diese Wörter in einem Dataframe in einer Reihe stehen, werden diese 150 Wörter auf 150 Reihen aufgeteilt.

Titel	Inhalt	Datum
Titel eines Artikels	Inhalt eines Artikels	Datum



Titel	Wort	Datum
Titel eines Artikels	inhalt	Datum
Titel eines Artikels	eines	Datum
Titel eines Artikels	artikels	Datum

Abbildung 3-17: Tokenisierung (Quelle: eigene Abbildung)

Die Tokenisierung teilt den Inhalt nicht nur auf, sie wandelt die Wörter auch in Kleinschreibung um. Diese Kleinschreibung muss beim Verschmelzen mit dem Sentiment Lexikon beachtet werden.

```
sentiment_per_article_table = reactive({
    temp = filtered_data_library()
    temp = temp[as.Date(temp$date) > as.Date(input$v_date)-10 &
      as.Date(temp$date) <= as.Date(input$v_date),]

    unnested_df = temp %>%
      unnest_tokens(word, content)

    germanlex$word = tolower(germanlex$word)
    adjustGermanlex = germanlex
    adjustGermanlex$sentiment = ifelse(germanlex$qualifier == 'NEG',  -
germanlex$polarity_strength,
                                       ifelse(germanlex$qualifier ==
'POS', germanlex$polarity_strength, 0))

    test = unnested_df %>%
      inner_join(adjustGermanlex) %>%
      group_by(title)
```

Listing 3-20: Tokenisieren und mit Sentiment Lexikon verschmelzen

Durch die Funktion unnest_tokens des R-Pakets „tidytext“ findet die Tokenisierung statt. Um das Sentiment einfacher bestimmen zu können, wird das Dataframe *germanlex* von dem R-

Paket „pradadata“ um eine Spalte erweitert. In dieser Spalte wird die Stärke der Polarität (polarity_strength) mit der Qualifikation (qualifier) abgestimmt.

qualifier	polarity_strength
NEG	0.5
POS	0.7

qualifier	polarity_strength	sentiment
NEG	0.5	-0.5
POS	0.7	0.7

Abbildung 3-18: Sentiment Lexikon anpassung (Quelle: eigene Abbildung)

Mit dieser „sentiment“ Spalte wird das Addieren dieser Werte in der Auswertung erleichtert. Mit einem inner_join() des tokenisierten Dataframes und des angepassten Sentiment Lexikon, werden diese Dataframes vereint und weiters nach Titel gruppiert, um jeweils die Summe des Sentiments für diesen Artikel zu bekommen.

```
  date_sentiment_df = test[c("date", "sentiment")]
  date_sentiment_df$date = format(as.Date(date_sentiment_df$date,
    "%Y-%m-%d"))

  dta.sum <- aggregate(x = date_sentiment_df["sentiment"],
                       FUN = sum,
                       by = list(Group.date = date_sentiment_df$date))

  temp$date = format(as.Date(temp$date, "%Y-%m-%d"))
  article_count = temp %>%
    count(date)

  article_sent_count = inner_join(dta.sum, article_count, by =
    c("Group.date" = "date"))
  article_sent_count$sentperarticle = article_sent_count$sentiment /
    article_sent_count$n
  article_sent_count
})
```

Listing 3-21: Sentiment pro Artikel

Um bestimmen zu können, wann ein Sentiment-Wert einen positiven oder negativen Effekt haben kann, muss zuerst eine Basis gefunden werden. Diese Basis wird über einen Mittelwert definiert. Der Mittelwert soll aus dem Sentiment pro Artikel entstehen. Hierfür muss das bis jetzt generierte Dataframe nach Datum gruppiert werden. So entsteht ein kumulierter Sentiment-Wert pro Tag. Da es eine unterschiedliche Anzahl an Artikel pro Tag gibt, ist der Sentiment-Wert pro Tag kein guter Referenzwert. Somit wird die Anzahl an Artikeln pro Tag berechnet. Wird nun der Sentiment-Wert pro Tag durch die Anzahl an Artikeln gerechnet, entsteht ein aussagekräftiger Wert. Aus diesen Werten wird ein Mittelwert erstellt, um erkennen zu können,

ob die Anomalie beziehungsweise der selektierte Tag, einen positiven oder negativen Einfluss auf den ausgewählten Themenbereich haben kann.

```
output$sentimenttimeseries = renderPlot({
  df = sentiment_per_article_table()
  mittelwert = mean(df$sentperarticle)
  ggplot(df, aes(x = Group.date, y = sentperarticle))+
    geom_point() +
    geom_hline(yintercept = mittelwert, col = "blue")
})
```

Listing 3-22: Diagramm Sentiment Analyse mit Mittelwert

Diese entstandene Tabelle und der ausgerechnete Mittelwert, kann in einem Scatterplot für eine visuelle Unterstützung dargestellt werden und um zusätzlich die Entfernung des Mittelwerts zu sehen. Höhere Entfernung zum Mittelwert kann in größerem positivem oder größerem negativem Einfluss resultieren.

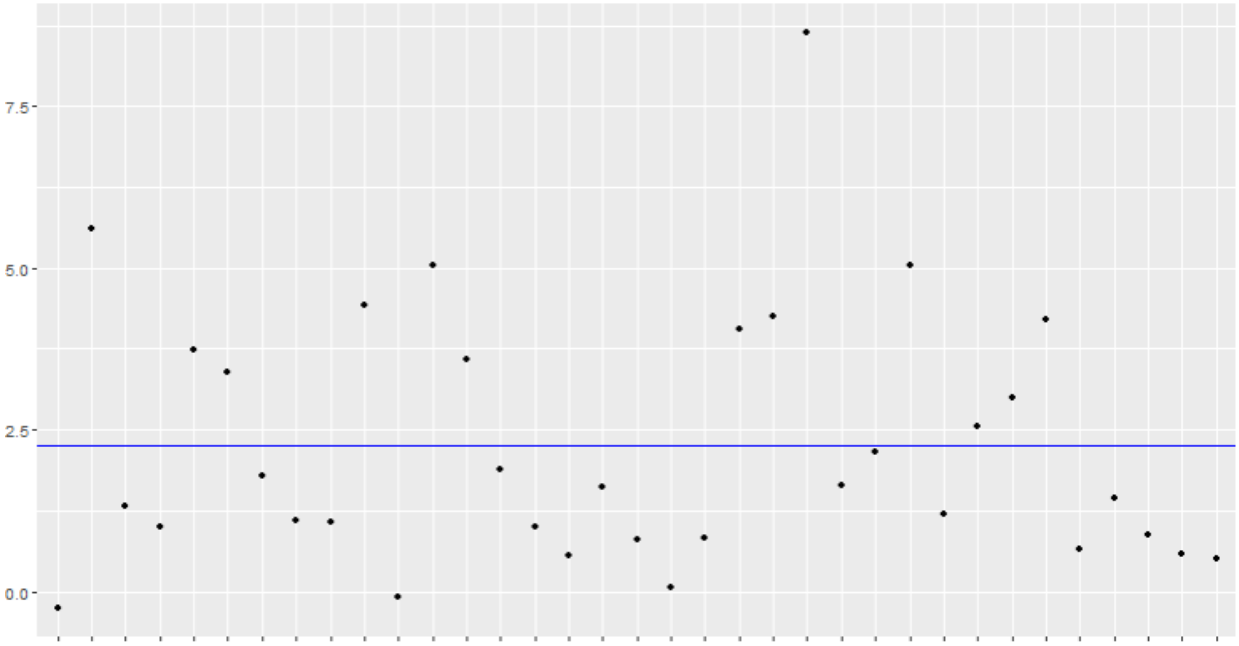


Abbildung 3-19: Diagramm Sentiment Analyse mit Mittelwert (Quelle: eigene Abbildung)

Ein weiterer Schritt, welcher mit dieser Analyse durgeführt werden kann, ist der Vergleich mit der Zeitreihenanalyse. Es kann überprüft werden, ob Gemeinsamkeiten auftreten. Resultiert eine größere Anzahl an Vorkommen in einem größeren Sentiment-Wert pro Artikel?

3.4.7 Kookkurrenz

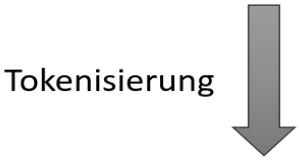
Für eine weitere Auswertung der Daten und eine mögliche Interpretation dessen wird eine Kookkurrenz Analyse durchgeführt. Die Kookkurrenz Analyse kann wichtige Muster in der Verwendung von Wortkombinationen im Text erkennen. Weiters liefert sie Hinweise auf systematisches Vorkommen von Wörtern und ein Maß für deren Affinität (Schnörch, 2019) (Leibnitz-Institut für deutsche Sprache, kein Datum).
Da für dieses Vorgehensmodell die Kookkurrenz des Tages der Anomalie beziehungsweise des ausgewählten Tages ist, wird ein Datensatz benötigt, der nach Themenbereich und Tag vorselektiert ist. Dieser Datensatz wird für die Kookkurrenz „tokenisiert“. Es gibt verschiedene Arten von Tokenisierung. Es kann jedes Wort einzeln extrahiert werden, aber es kann auch ein Satz als Token deklariert werden. Für die Kookkurrenz müssen Wortpaarungen von mindestens zwei Wörtern verwendet werden. Ein wichtiger Baustein für eine Aussagekräftige Kookkurrenz ist das Entfernen von irrelevanten Wörtern, auch Stoppwörter genannt. Um die Stoppwörter auszusortieren, wird das R-Paket „stopwords“ verwendet.

```
friends_bigram <- day_and_keyword_filtered() %>%
  unnest_tokens(word, content, token = "ngrams", n = 2) %>%
  separate(word, into = c("word1", "word2"), sep = " ") %>%
  filter(!word1 %in% stopwords("german"),
         !word2 %in% stopwords("german")) %>%
  filter(! (is.na(word1) | is.na(word2)))
```

Listing 3-23: Zwei Wort Tokenisierung und Stoppwort Filter

Die Zwei Wort Tokenisierung funktioniert wie folgt.

Titel	Inhalt	Datum
Titel eines Artikels	Inhalt eines wichtigen Artikels	Datum



Titel	Wort1	Wort2	Datum
Titel eines Artikels	inhalt	eines	Datum
Titel eines Artikels	eines	wichtigen	Datum
Titel eines Artikels	wichtigen	artikels	Datum
Titel eines Artikels	artikels		Datum

Abbildung 3-20: Zwei Wort Tokenisierung (Quelle: eigene Abbildung)

Aus diesem tokenisierten Dataframe werden die gemeinsamen Vorkommen gezählt und ein neues Dataframe erstellt.

```
friends_bicount <- friends_bigram %>%
  count(word1, word2, sort = TRUE)
```

Listing 3-24: kookkurrierende Wörter zählen

Da dieser Datensatz nicht leicht interpretierbar ist, da immer nur zwei Wortpaarungen betrachtet werden, wird dieser Datensatz in einer Wortwolke dargestellt, um weitere Abhängigkeiten zwischen Wörtern anzeigen zu können. Dies bietet mit der Sentiment Analyse und der Zeitreihenanalyse eine weitere Möglichkeit zur Interpretation. Weiters können die Wörter der Wortwolke mit überprüft werden, ob sie mit dem Wert der Sentiment Analyse Berührungspunkte haben.

```
friends_bicount %>%
  filter(n > 1) %>%
  as_tbl_graph() %>%
  ggraph(layout = "fr") +
  geom_edge_link(aes(alpha = n), show.legend = FALSE,
                 end_cap = circle(0.07, "inches")) +
  geom_node_point(color = "lightblue", size = 5) +
  geom_node_text(aes(label = name), size = 4, vjust = 1, hjust = 1)
```

Listing 3-25: Wortwolken-Diagramm

Da für die Wortwolke nicht alle Wortpaarungen genommen werden, wird ein Filter festgelegt. In diesem Beispiel werden alle Wortpaarungen genommen, welche öfter als einmal auftreten. Dieser Wert kann und muss für größere Datenmengen angepasst werden, da Wortpaarungen öfter auftreten. In der folgenden Abbildung werden öfter Auftretende Wortpaarungen durch einen stärkeren schwarzen Strich angezeigt.

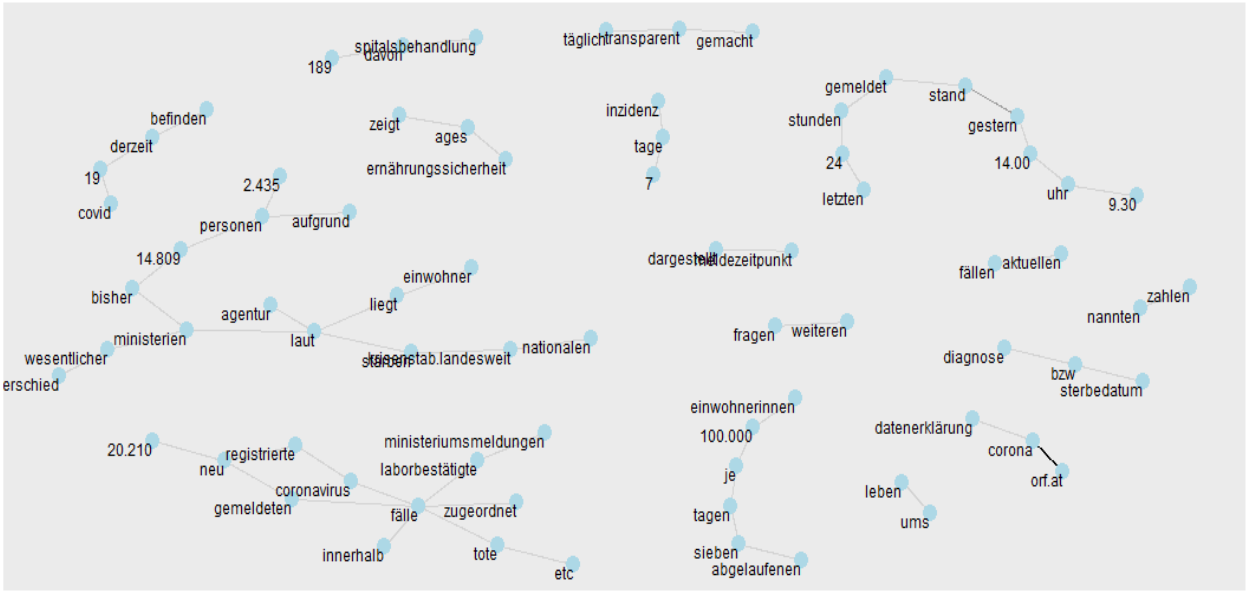


Abbildung 3-21: Wortwolken-Diagramm (Quelle: eigene Abbildung)

4 ERGEBNISSE

Die Ergebnisse werden mit Hilfe eines Fallbeispiels erzeugt. In diesem Fallbeispiel soll anhand einer Problemstellung die Aussagekraft des Vorgehensmodells beziehungsweise des Prototyps überprüft werden, um Ergebnisse für die Beantwortung der Forschungsfrage zu erhalten.

4.1 Mögliche Problemstellung

Eine mögliche Problemstellung, welche mit Hilfe des Vorgehensmodell gelöst werden soll, könnte wie folgt aussehen. Man geht von einer globalen Pandemie (Covid19) aus, welche verschiedene Auswirkungen auf eine Firma haben kann. Diese Firma hängt sehr am Import diverserer Produkte. Das könnten Anlagen aus Frankreich, Elektronikbauteile aus China, oder Chemikalien aus Niger sein.

Aufgrund dieser Tatsachen ist es nun essenziell eine Anomalie zu erkennen, und ob diese in diesen Regionen auftritt und wie sich diese Anomalie auswirken kann. Wird durch eine negative Anomalie in diesen Regionen die Lieferzeit für die Import Ware erhöht, da es eine Steigerung an krankheitsbedingten Ausfällen gibt oder können Lieferzeiten durch eine positive Anomalie gesenkt werden. Welche Interpretation lässt das Ergebnis des Vorgehensmodells zu?

4.2 Datensatz

Die Auswahl des Datensatzes fiel auf den Nachrichten Datensatz, da Reddit und Twitter Daten viele irrelevanten Daten haben und für dieses Themengebiet nicht brauchbar sind.

Der Datensatz besteht aus etwa 3000 Nachrichten Artikel der Plattform „orf.at“.

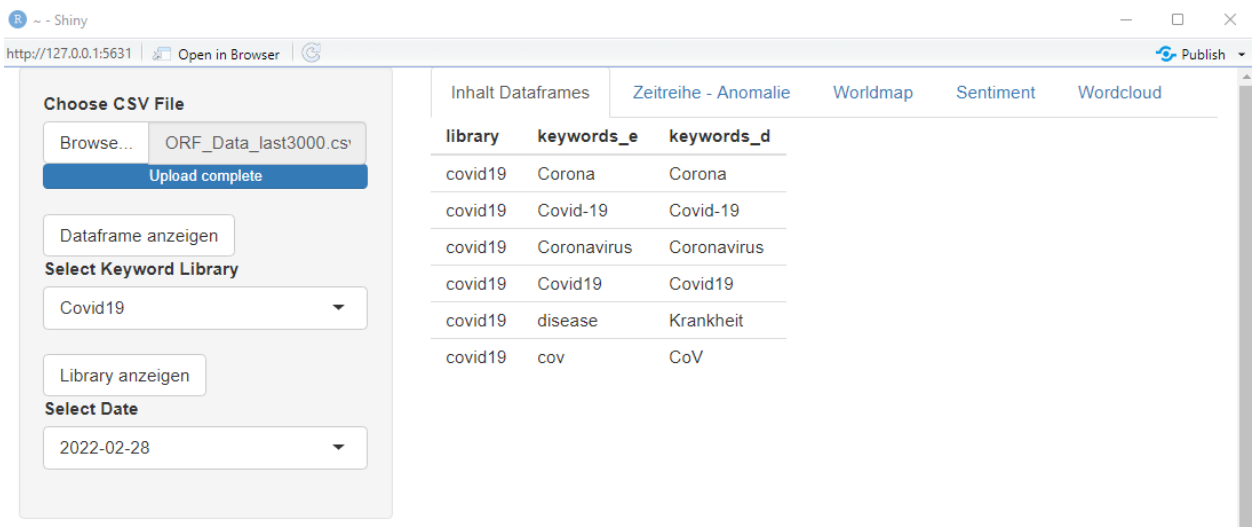


Abbildung 4-1: Datensatz geladen + Themenbereich Ausgabe (Quelle: Eigene Abbildung)

Direkt nachdem der Datensatz geladen ist, soll ein Pop-Up Fenster erscheinen, wenn eine Anomalie im Datensatz für den ausgewählten Tag erkannt wird. In diesem Datensatz wird keine Anomalie erkannt. Doch es gibt noch weitere Analysen dieses Datensatzes, welche von Bedeutung sein können. Die geladene Tabelle in dieser Abbildung, zeigt an, nach welchen Schlüsselwörtern in diesem Themenbereich gefiltert wird.

4.2.1 Zeitreihen

Die Zeitreihe zeigt an, wie die Anzahl an gefundenen Artikeln in Bezug auf den Themenbereich ist. Falls die Grenze am aktuellsten Tag überschritten wird, wird der Punkt auf der Zeitreihe Rot angezeigt und es würde beim Laden des Datensatzes ein Pop-Up Fenster geöffnet.

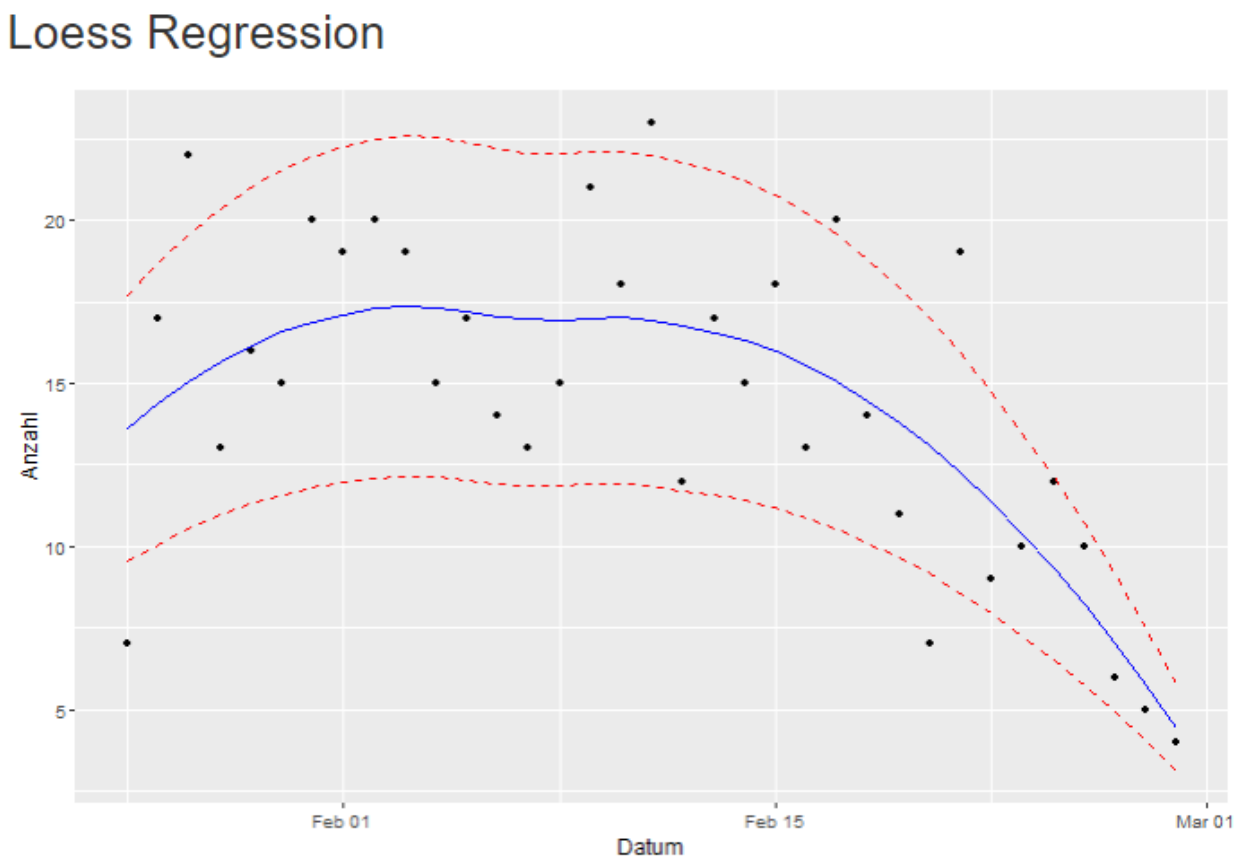


Abbildung 4-2: LOESS Zeitreihe (Quelle: eigene Abbildung)

Es ist erkennbar, dass der Wert innerhalb der definierten Grenzen liegt und somit nicht als Anomalie gekennzeichnet wird. Aus diesem Diagramm ist auch erkennbar, dass die Vorkommen für den Themenbereich Covid19 Anfang Februar bis Mitte Februar höher sind als zum Ende des Monats.

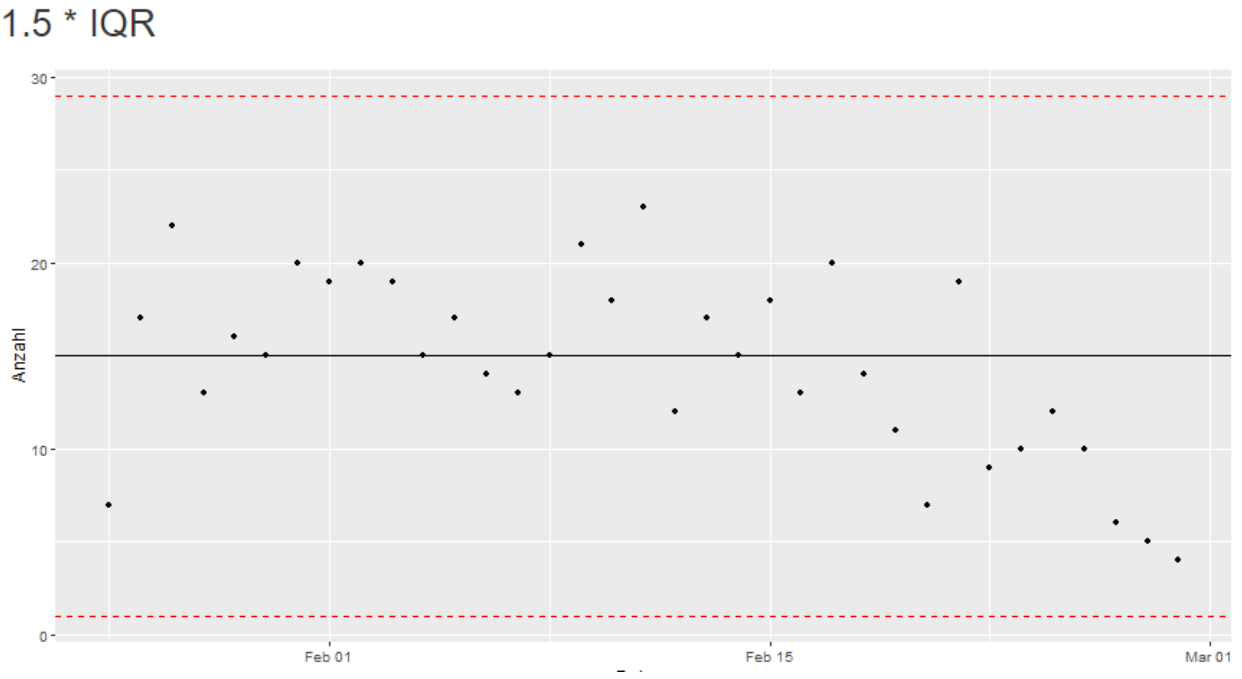
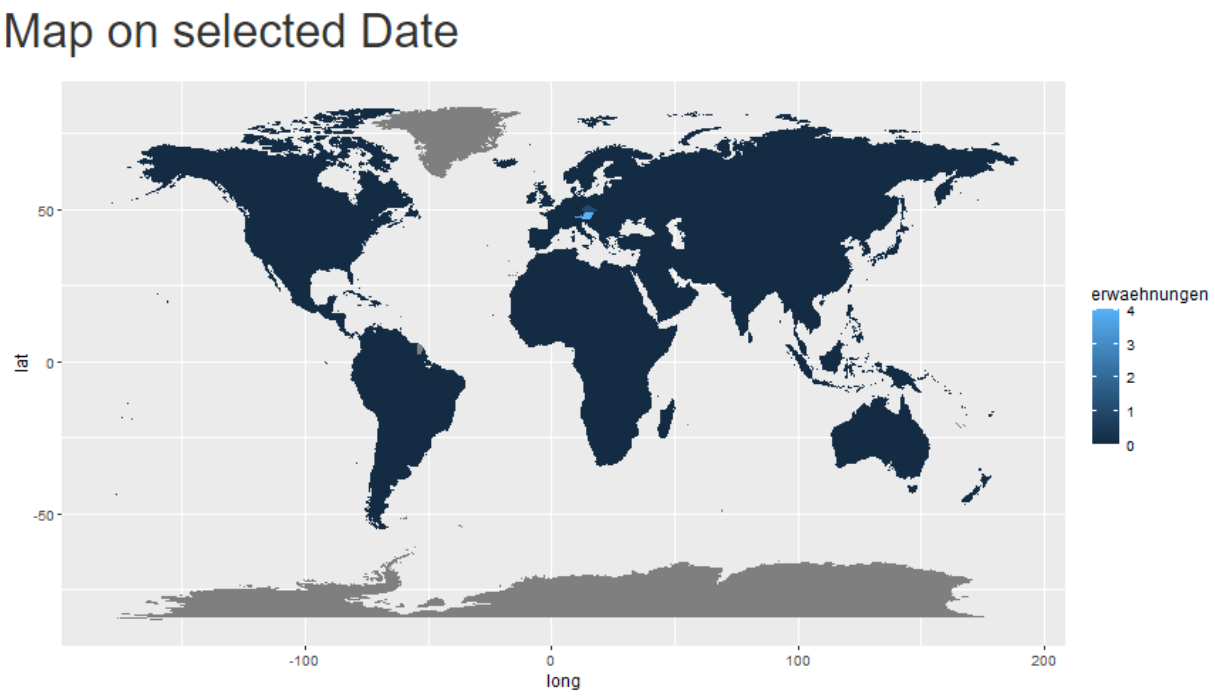


Abbildung 4-3: IQR Zeitreihe (Quelle: eigene Abbildung)

Auch bei der zweiten Methode zur Anomalie Erkennung, wird keine Grenze überschritten. Da alle Werte weit von den Grenzen entfernt sind, muss überlegt werden, ob die Grenzen für diesen Datensatz und diesen Themenbereich neu gesetzt werden müssen.

4.2.2 Zentren in Daten erkennen – Weltkarte

Können Zentren beziehungsweise Hotspots auf einem Wärmebild erkannt werden, welche Aufschlüsse für Risiken oder Chancen geben?



region	erwaehnungen
Austria	4.00
Czech Republic	1.00
Abkhazia	0.00
Afghanistan	0.00
Egypt	0.00

Abbildung 4-4: Weltkarte und Wärmebild (Quelle: eigene Abbildung)

Nein, in diesem Datensatz nicht. Das könnte mitunter den Grund haben, dass es an diesem Tag nicht viele Artikel bezügliche Covid19 gegeben hat. Der Wert von Österreich (Austria) in der Tabelle unter der Weltkarte und das Aufleuchten in der Weltkarte muss mit Bedacht interpretiert werden, da die Daten von einer österreichischen Nachrichten Plattform stammen.

4.2.3 Sentiment

Da das Sentiment pro Artikel berechnet wird, hat die Anzahl an Artikeln für diese Analyse keine signifikante Relevanz. Wie kann das Sentiment dieses Datensatzes nach dem Vorgehensmodell interpretiert werden?

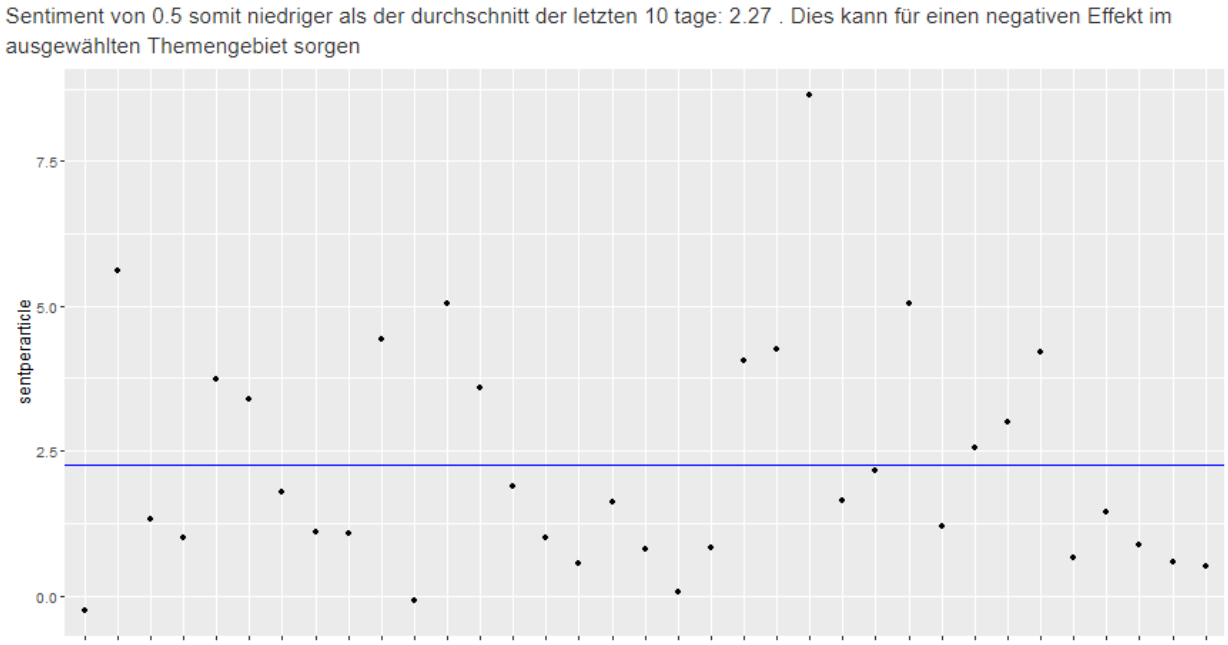


Abbildung 4-5: Sentiment Analyse (Quelle: eigene Abbildung)

Das Sentiment des ausgewählten Tages liegt unter dem Durchschnitt der betrachteten Daten, was auf einen negativen Effekt deuten kann. Könnte heißen, dass die Rate Erkrankungen erhöht ist, oder das Gesundheitswesen überlastet ist.

4.2.4 Wortwolke

Es wird in den Daten zwar keine Anomalie erkannt, aber der Sentiment-Wert liegt weit unterhalb des Durchschnitts. Kann die Wortwolke weitere Daten für die Interpretation liefern?

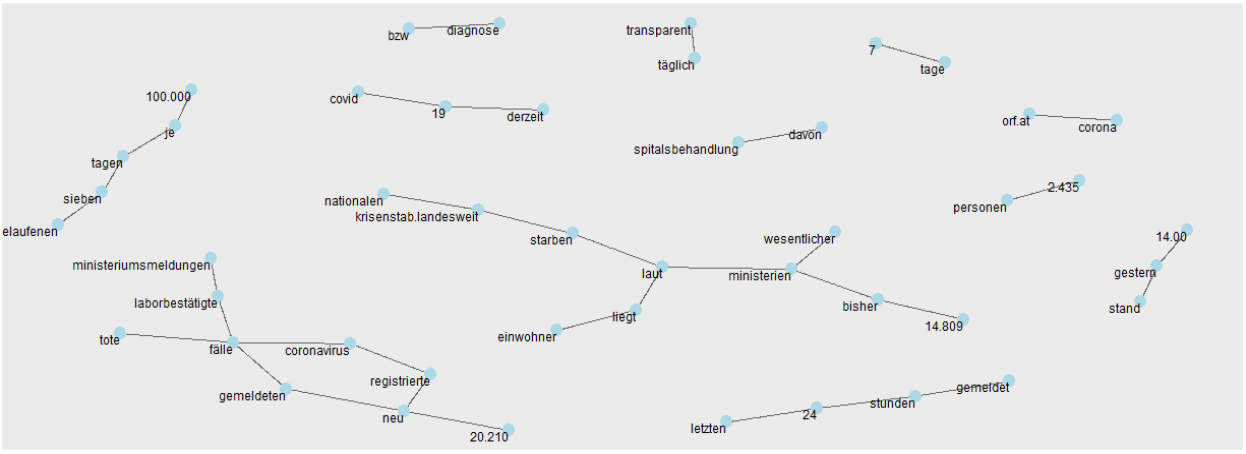


Abbildung 4-6: Wortwolke (Quelle: eigene Abbildung)

Bei genauer Betrachtung der Wortwolke können Rückschlüsse auf das negative Sentiment gezogen werden. Es tauchen Wortkombinationen auf wie:

- davon – spitalsbehandlung
- nationalen – krisenstab.landesweit – starben
- neu – gemeldeten – fälle – tote

4.2.5 Ergebnis

Für die Beantwortung der Problemstellung kann dieser Datensatz wie folgt interpretiert werden:

Es wird keine Anomalie in der Zeitreihenanalyse erkannt, jedoch ist eine Trendentwicklung erkennbar. Der Themenbereich kommt immer seltener vor. Diese Tatsache kann auch anderen Ereignissen geschuldet sein (Ukraine Krieg). Bei der Weltkarte kann auch keine Region identifiziert werden, welche dieses Themengebiet an diesem Tag betrifft. Die Sentiment Analyse zeigt einen Wert unter dem Durchschnitt, welcher auf eine Verschlechterung in diesem Themenbereich schließen lässt. Dieses negative Sentiment spiegelt die Kookkurrenz in der Wortwolke wider.

4.3 Datensatz modifiziert

Um eine künstliche Anomalie in einem Datensatz zu erzeugen, wird der bestehende Datensatz genommen und modifiziert. Für die Modifikation werden neue Artikel in diesem Datensatz geschrieben auf welche der Prototyp reagieren kann.

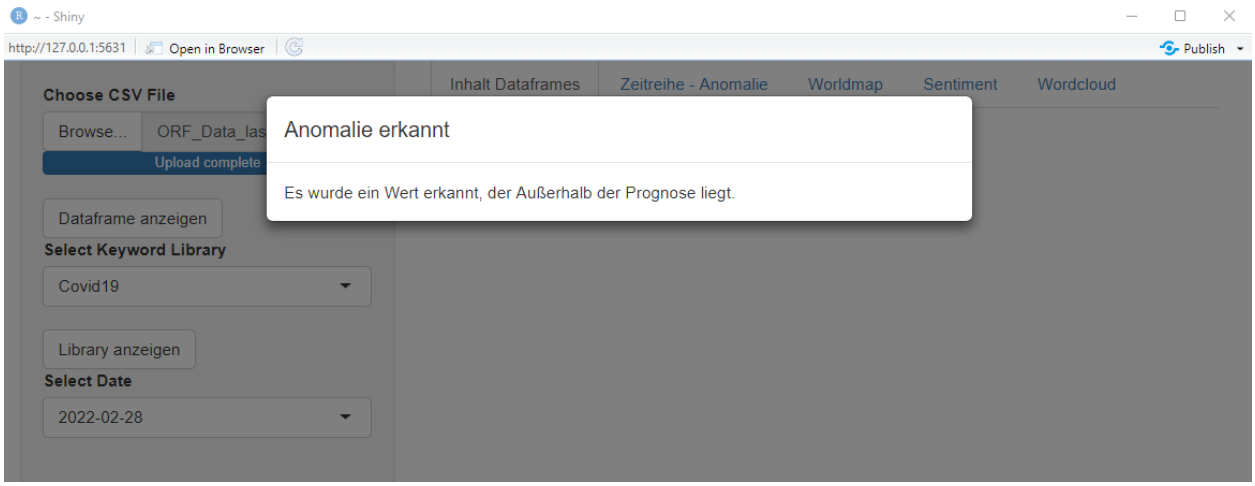


Abbildung 4-7: Anomalie erkannt Pop-Up (Quelle: eigene Abbildung)

Beim Laden des modifizierten Datensatzes, wird eine Anomalie erkannt und es erscheint ein Pop-Up Fenster, dass der Benutzerin / dem Benutzer signalisiert, dass es mögliche Einflüsse auf den Themenbereich gibt.

4.3.1 Zeitreihen

Im Zeitreihen Diagramm wird die Anomalie rot hervorgehoben. Da sie über den Grenzen der LOESS Kurve ist, gibt es an diesem Tag mehr Artikel in diesem Themenbereich, als die LOESS Funktion errechnet hat.

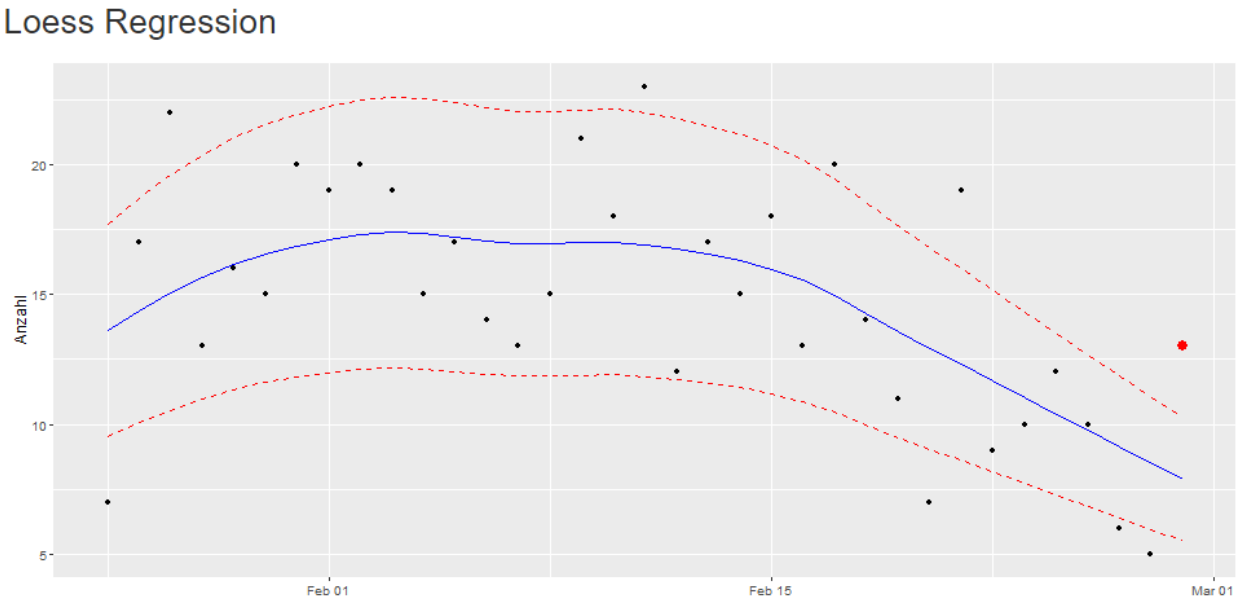


Abbildung 4-8: LOESS Zeitreihe modifiziert (Quelle: eigene Abbildung)

Im Gegensatz zur Zeitreihen Analyse mit der LOESS Funktion, wird bei der IQR-Methode kein Ausreißer erkannt. Bei dieser Anomalie Erkennung liegt der Wert sogar ziemlich im Mittel.

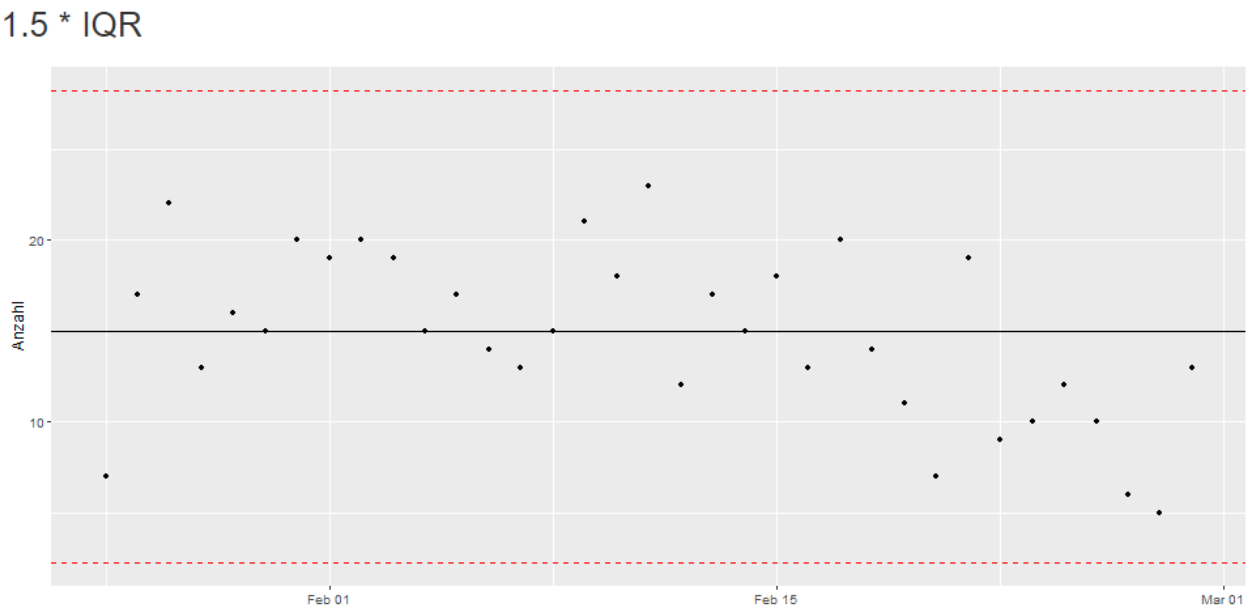
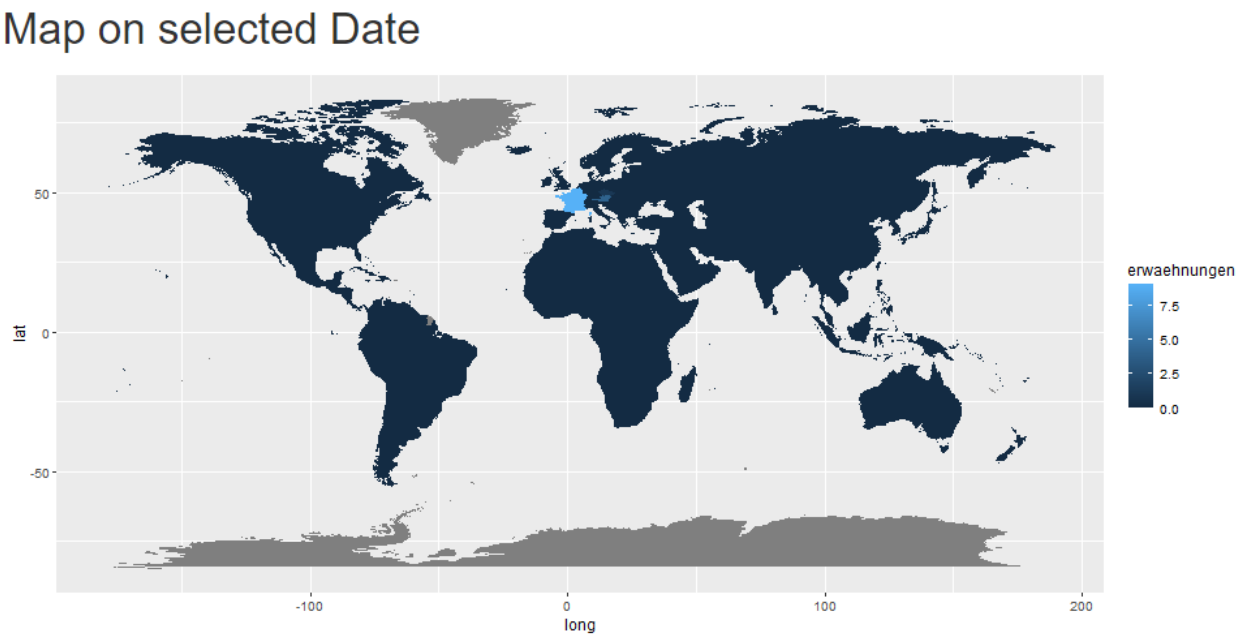


Abbildung 4-9: IQR Zeitreihe modifiziert (Quelle: eigene Abbildung)

4.3.2 Zentren in Daten erkennen – Weltkarte

Können Zentren beziehungsweise Hotspots in dem modifizierten Datensatz erkannt werden?

Ja. Es gibt einen Hotspot in Frankreich und Belgien. In diesen Regionen ist der Themenbereich sehr akut. Ob dieser Hotspot positive oder negative Effekte hat, muss in der Sentiment Analyse und der Wortwolke eruiert werden.



region	erwaehnungen
Belgium	9.00
France	9.00
Austria	4.00
Czech Republic	1.00
Abkhazia	0.00

Abbildung 4-10: Weltkarte und Wärmebild modifiziert (Quelle: eigene Abbildung)

4.3.3 Sentiment

Der Sentiment-Wert ist weit über dem Durchschnitt was für positive Effekte in den Regionen Frankreich und Belgien sorgen kann.

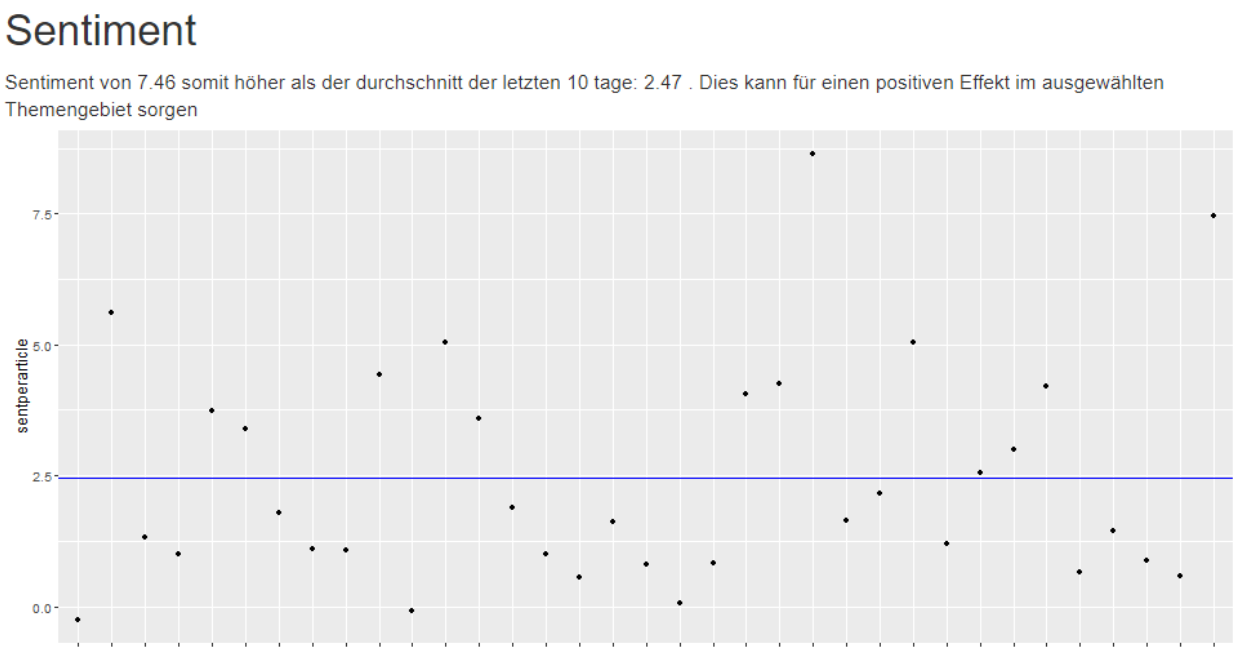


Abbildung 4-11: Sentiment Analyse modifiziert (Quelle: eigene Abbildung)

4.3.4 Wortwolke

Es wird in den Daten bereits eine Anomalie erkannt und der Sentiment-Wert ist weit über dem Durchschnitt. Kann die Wortwolke weitere Daten für die Interpretation liefern?

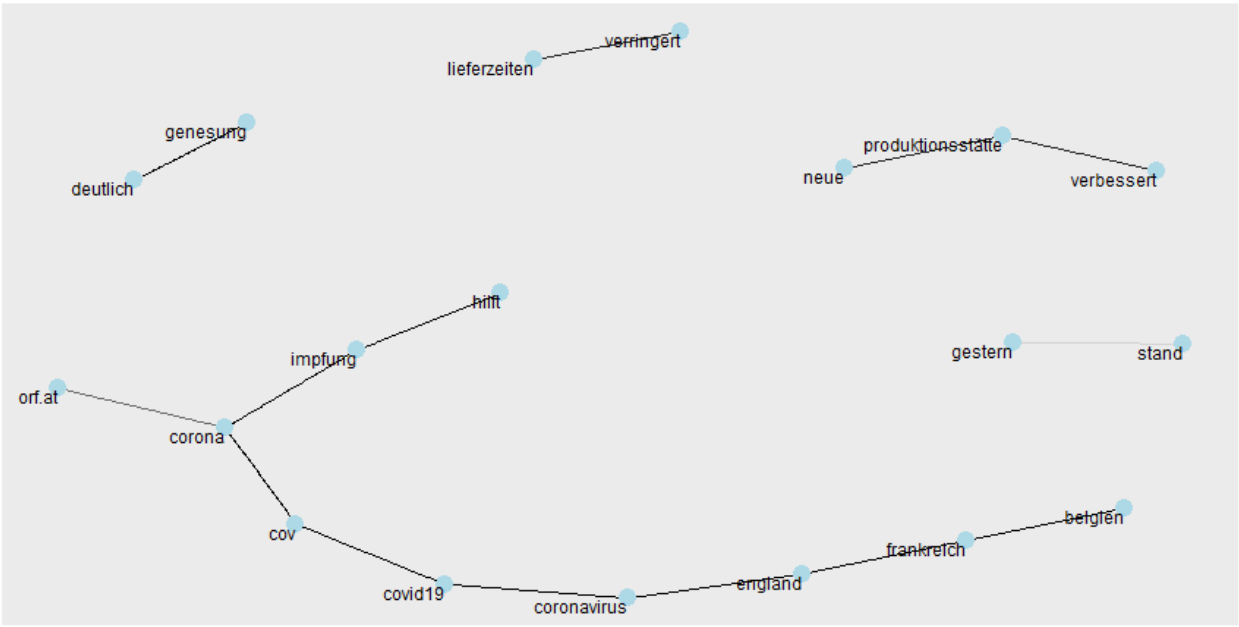


Abbildung 4-12: Wortwolke modifiziert (Quelle: eigene Abbildung)

Ja kann sie. Bei der Betrachtung der Wortwolke kann auch der Grund für das positive Sentiment erkannt werden. Es entstehen Wortpaarungen wie:

- lieferzeiten – verringert

- corona – impfung – hilft
- neue – produktionsstätte – verbessert
- genesung – deutlich

4.3.5 Ergebnis

Für die Beantwortung der Problemstellung kann dieser Datensatz wie folgt interpretiert werden:

Es wird eine Anomalie in der Zeitreihenanalyse erkannt, welche gegen den Trend spricht. Diese Anomalie kann auf Regionen bezogen werden. Da der Sentiment-Wert positiv ist, kann es einen positiven Einfluss in diesen Regionen in Hinsicht auf den Themenbereich geben und durch die Kookkurrenz kann interpretiert werden, dass es in diesen Regionen zu neuen Produktionsstätten kommt. Diese Produktionsstätten können hilfreich für die Lieferzeiten der Problemstellung sein.

Um genauere Aussagen treffen zu können, ob die Produktionsstätten für den Anlagenbau verwendet werden, muss eine Anpassung der Schlüsselwörter in den Themenbereichen vorgenommen werden.

4.4 Fazit

Das Vorgehensmodell funktioniert und kann auf jeden Datensatz angewendet werden, wenn die richtige Dataframe Struktur vorliegt. Durch das Zusammenspiel der Analysen im Vorgehensmodell entstehen Möglichkeiten zur Interpretation, ohne dass hunderte von Artikeln gelesen werden müssen.

Der wichtigste Baustein für diese Interpretationsmöglichkeiten liegt in den Schlüsselwörtern des Themenbereichs. Je besser diese spezifiziert sind, desto genauer können in digitalen Informationsströmen Anomalien erkannt und über ihre Semantik ausgewertet werden. Weiters ist bei der Spezifikation der Schlüsselwörter zu beachten, dass sich diese im Laufe der Zeit ändern können.

5 ZUSAMMENFASSUNG

Diese Arbeit befasst sich mit der Fragestellung *„Welche Möglichkeiten zur Interpretation und Bewertung ergeben sich durch die Identifikation von Anomalien und deren Semantik in Informationsflüssen?"*

Zur Beantwortung dieser Frage wurde ein Vorgehensmodell erstellt, welches die Basis für einen Prototyp lieferte. Für das Vorgehensmodell wurde zunächst eine Umfangreiche Literaturrecherche gemacht, um die richtigen Analysen und Verfahren für das Vorgehensmodell auszuwählen.

Über den Prototyp können verschiedene Textdatensätze eingelesen werden, um nicht nur von einer Datenquelle abhängig zu sein. Diese eingelesenen Daten werden automatisch gefiltert, analysiert und meist als Tabelle oder Diagramm dargestellt. Durch die Analyse der Semantik der Textdaten entstehen Interpretationsmöglichkeiten wie: „Es wurde eine erhöhte Anzahl an Artikeln in diesem Themenbereich entdeckt“, oder „der Artikel hat einen positiven Sentiment-Wert, deshalb kann er einen positiven Einfluss auf einen Themenbereich haben“.

Der Themenbereich ist für den Prototypen der wichtigste Baustein, da die Schlüsselwörter des Themenbereichs durch die Filterung des Datensatzes erst die Anomalie Erkennung ermöglichen.

Durch eine weitere Filtermethode nach Ländernamen beziehungsweise deren Hauptstädten, wird eine Weltkarte mit Wärmebild Funktion erstellt, um Zentren / Hotspots auf der Weltkarte für diesen Themenbereich zu eruieren.

Zum Schluss des Prototyps wird eine Kookkurrenz Analyse durchgeführt, um Zusammenhängende Wörter identifizieren zu können. Diese werden für einen besseren Überblick und bessere Verständlichkeit als Wortwolke grafisch dargestellt.

Das Zusammenspiel aller Analysen grenzt die Möglichkeiten zur Interpretation und Bewertung von Anomalien und deren Semantik ein.

5.1 Ausblick

Der Prototyp ermöglich die Anomalie Erkennung verschiedene Analysen durchführen, jedoch müssen für verschiedene Themenbereiche dynamische Funktionen implementiert werden, um zum Beispiel die Grenzen anpassen zu können.

Eine wichtige Implementierung könnte auch sein, dass die Themenbereiche in der Anwendung direkt editiert werden können, um die Filterung anzupassen.

Interessant wäre auch ein automatischer Vergleich zwischen der Zeitreihe und den Sentiment-Werten. Gibt es Korrelationen zwischen diesen Werten?

Ein großes Forschungsgebiet, für welches diese Arbeit relevant sein kann, wäre Deep Learning / Machine Learning. Die analysierten Daten des Prototyps könnten täglich extrahiert und gesammelt werden, um durch maschinelles Lernen Zusammenhänge erfassen zu können und so Prognosen zu bekommen.

ABBILDUNGSVERZEICHNIS

TABELLENVERZEICHNIS

LISTINGS

LITERATURVERZEICHNIS

(2014). Von Reddit: https://www.reddit.com/r/Enhancement/comments/1zoly3/does_reddit_limit_the_depth_of_comment_nesting_at/ abgerufen

Benoit, K., Watanabe, K., Wang, H., Nulty, P., Obeng, A., Müller, S., & Matsuo, A. (2018). quanteda: An R package for the quantitative analysis of textual data. In *Journal of Open Source Software.* doi:10.21105/joss.00774

Bouveyron, C., Celeux, G., Murphy, T. B., & Raftery, A. E. (2019). *Model-Based Clustering and Classification for Data Science: With Applications in R.* Cambridge University Press.

Buhl, Y. (29. 06 2019). *Package 'newsanchor'.* Von https://cran.r-project.org/web/packages/newsanchor/newsanchor.pdf abgerufen

Cambria, E., Das, D., Bandyopadhyay, S., & Feraco, A. (2017). *A Practical Guide to Sentiment Analysis.* Springer International Publishing AG.

Chaudhary, S. (28. September 2019). *towardsdatascience.com*. Von https://towardsdatascience.com/why-1-5-in-iqr-method-of-outlier-detection-5d07fdc82097 abgerufen

Feldman, R., & Sanger, J. (2007). *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data.* New York: Cambridge University Press.

internet live stats. (2022). Von https://www.internetlivestats.com/twitter-statistics/ abgerufen

Kirchgässner, G., & Wolters, J. (2007). *Introduction to Modern Time Series Analysis.* Berlin, Heidelberg, New York: Springer.

Kumar, A., & Paul, A. (2016). *Mastering Text Mining with R.* Packt.

Kwartler, T. (2017). *Text Mining in Practice with R.* John Wiley & Sons .

Leibnitz-Institut für deutsche Sprache. (kein Datum). *Leibniz-Institut für deutsche Sprache*. Von https://www.ids-mannheim.de/digspra/kl/projekte/methoden/ka/ abgerufen

Mehrotra, K., Mohan, C., & Huang, H. (2017). *Anomaly Detection Principles and Algorithms.* Springer International Publishing AG.

Munzert, S., Rubba, C., Meißner, P., & Nyhuis, D. (2015). *Automated Data Collection with R.* John Wiley & Sons, Ltd.

NewsAPI. (kein Datum). *NewsAPI*. Von https://newsapi.org/docs/endpoints/everything abgerufen

NIST/SEMATECH. (30. 10 2013). e-Handbook of Statistical Methods. doi:https://doi.org/10.18434/M32189

Sachs-Hombach, K., & Zywietz, B. (2018). *Fake News, Hashtags & Social Bots.* Springer Fachmedien Wiesbaden GmbH.

Schmuller, J. (2017). *Statistical Analysis with R For Dummies.* New Jersey: John Wiley & Sons, Inc.

Schnörch, U. (2019). *Kookkurrenzanalyse und Vergleich: Überlegungen zur Methodenanwendung bei der lexikografischen Beschreibung von Paronymen.* Mannheim: Institut fpr Deutsche Sprache Mannheim.

Schulze, L. (29. 06 2019). *Usage of newsanchor*. Von https://cran.r-project.org/web/packages/newsanchor/vignettes/usage-newsanchor.html abgerufen

Silge, J., & Robinson, D. (2017). *Text Mining with R A Tidy Approach.* O'Reilly Media.

Stubblebine, T. (2007). *Regular Expression Pocket Reference, Second Edition.* Sebastopol, CA 95472: O'Reilly Media, Inc.

Suri, N., Murty M, N., & Athithan, G. (kein Datum). *Outlier Detection: Techniques and Applications.* 2019: Springer.

Twitter. (2021). *Developer Plattform Twitter*. Von https://developer.twitter.com/en/docs abgerufen

Weiss, S., Nitin, I., Zhang, T., & Damerau, F. (2005). *Text Mining: Predictive Methods for Analyzing unstructured Information.* New York: Springer.

Wickham, H. (2015). *R Packages.* O'Reilly Media.

Wickham, H. (2021). *Mastering Shiny.* O'Reilly Media.

Wickham, H., & Grolemund, G. (2017). *R for Data Science.* O'Reilly.

Zong, C., Xia, R., & Zhang, J. (2021). *Text Data Mining.* Springer.