

DIPLOMARBEIT

AI Interpretation von Börsennews

im Projekt AI Börse

Ausgeführt im Schuljahr 2020/21 von:

Alen Asanov, 5CHIF-01
Simon Mader, 5CHIF-11
Nicolas Philipp, 5CHIF-14
Louis Raschbach, 5CHIF-16

Betreuer/Betreuerin:

AV RR. Ing. Mag. Klaus Hasenzagl

St. Pölten, am 31. März 2021

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Diplomarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche erkenntlich gemacht habe.

Die Unterschrift ist in der Druckfassung ersichtlich.

Alen Asanov

Die Unterschrift ist in der Druckfassung ersichtlich.

Simon Mader

Die Unterschrift ist in der Druckfassung ersichtlich.

Nicolas Philipp

Die Unterschrift ist in der Druckfassung ersichtlich.

Louis Raschbach

St. Pölten, am 31. März 2021

Diplomandenvorstellung



Alen ASANOV

Geburtsdaten:
22.11.2001 in St. Pölten

Wohnhaft in:
Lehrnergasse 2
3452 Heiligenreich

Werdegang:
2016 - 2021:
HTBLuVA St. Pölten, Abteilung für Informatik
2012 - 2016:
NNÖMS Atzenbrugg

Kontakt:
a.asanov@htlstp.at

Diplomandenvorstellung



Simon MADER

Geburtsdaten:
17.02.2002 in St. Pölten

Wohnhaft in:
Kasten 13
3072 Kasten bei Böheimkirchen

Werdegang:
2016 - 2021:
HTBLuVA St. Pölten, Abteilung für Informatik
2012 - 2016:
Bundesrealgymnasium St. Pölten, musikalischer Zweig

Kontakt:
simon.mader17@gmail.com

Diplomandenvorstellung



Nicolas PHILIPP

Geburtsdaten:
20.05.2002 in Tulln

Wohnhaft in:
Werthfeldstraße 50
3441 Baumgarten am Tullnerfeld

Werdegang:
2016 - 2021:
HTBLuVA St. Pölten, Abteilung für Informatik
2012 - 2016:
Bundesrealgymnasium Tulln

Kontakt:
nicolas.philipp@gmx.at

Diplomandenvorstellung



Louis RASCHBACH

Geburtsdaten:
03.04.2002 in St. Pölten

Wohnhaft in:
Mitterfeld 79
3072 Kasten bei Böheimkirchen

Werdegang:
2016 - 2021:
HTBLuVA St. Pölten, Abteilung für Informatik
2012 - 2016:
NNÖMS Böheimkirchen

Kontakt:
raschbach.louis@gmail.com

Danksagung - Alen Asanov

Ich möchte die Möglichkeit ergreifen, um ein paar unglaublich tollen Menschen meinen Dank auszusprechen. Zuallererst danke ich meiner Familie für ihren finanziellen Support und ihren guten Willen, mich die ganzen 5 Jahre in der HTL durch zu bringen. Ohne ihren Gedanken, mich soweit zu bringen wie möglich wäre ich wahrscheinlich nie 5 Jahre an eine berufsbildende höhere Schule gegangen.

Als Nächstes bedanke ich mich herzlich bei zwei Kameraden meiner Klasse namens Enis Fetai und Florian Vesely dafür, dass sie jederzeit für mich da waren, wenn ich sie gebraucht habe und mir geholfen haben, wenn es brenzlig wurde. Nebenbei sind sie mir in meiner Laufbahn in der HTL sehr ans Herz gewachsen und ich würde ohne sie wahrscheinlich nicht hier sitzen und gerade meine Danksagung verfassen.

Einen großen Dank will ich auch meiner Projektgruppe aussprechen, da sie mich aus eigenen Stücken aufs Boot AI Börse geholt haben und es sehr viel Spaß gemacht hat, sowohl mit ihnen am Projekt zu arbeiten als auch unsere Freizeit. Noch dazu danke ich Herrn AV RR. Ing. Mag. Klaus Hasenzagl, unserem Projektbetreuer, für die gute Betreuung unseres Projektteams, ohne ihn wären wir sehr verloren mit unserer Diplomarbeit gewesen, jedoch hat er uns den Weg gezeigt und konstruktive Kritik zu unseren Arbeiten geäußert, damit wir das bestmögliche aus unserer Arbeit herausbekommen können.

Zu guter Letzt danke ich Peter Koch und meiner Freundin für die unglaublich hilfreiche Unterstützung bei meiner Diplomarbeit. Ohne die beiden wäre meine Grammatik und Rechtschreibung auf Volksschulniveau und meine Arbeit nur einen Bruchteil so gut wie sie momentan ist.

Danksagung - Simon Mader

Zuerst möchte ich mich bei meiner Familie bedanken, die mich während meiner gesamten Schullaufbahn und beim Erstellen dieser Diplomarbeit unterstützt und motiviert hat.

Des Weiteren möchte ich einen besonderen Dank an Herrn AV RR. Ing. Mag. Klaus Hasenzagl aussprechen, der uns bei der Erstellung dieser Arbeit betreut und geholfen hat.

Außerdem gilt ein besonderer Dank meinem Vater, der meine Arbeit korrekturgelesen hat und mir bei der Formulierung mancher Sätze geholfen hat.

Zuletzt möchte ich meinen Freunden danken, die mich in schwierigen Zeiten immer unterstützt haben.

Danksagung - Nicolas Philipp

An erster Stelle möchte ich mich bei allen bedanken, die mich beim Schreiben der Diplomarbeit unterstützt haben.

Weiterhin danke ich unserem Diplomarbeitsbetreuer Herrn AV RR. Ing. Mag. Klaus Hasenzagl, welcher uns im Zuge der Arbeit sowohl bei projektbezogenen Fragen als auch bei diplomarbeitsbezogenen Fragen zur Verfügung stand.

Außerdem möchte ich mich bei meiner Tante für das Überprüfen der Diplomarbeit auf Grammatik- und Rechtschreibfehler und somit für den Beitrag zu einer fehlerfreien Arbeit bedanken. Ebenso gilt mein Dank meinen Eltern für das Korrekturlesen meiner Arbeit.

Zuletzt möchte ich meinen Gruppenmitgliedern danken. Durch die gegenseitige Unterstützung und Ergänzung hat die Entwicklung im Projekt sehr gut funktioniert.

Danksagung - Louis Raschbach

Ich möchte diese Gelegenheit nutzen, um allen Personen zu danken, die mich bei der Erstellung dieser Arbeit unterstützt haben.

Erstmals will ich meiner Mutter danken, die mich während meiner gesamten Schullaufbahn immer unterstützt hat und mir immer die notwendigen Mittel bereitgestellt hat, um alle Aufgaben bestens zu erfüllen.

Ein besonderes Dankeschön möchte ich an unseren Diplomarbeitsbetreuer, Herrn AV RR. Ing. Mag. Klaus Hasenzagl richten, der mir im Laufe unseres Projekts sowie der Verfassung meiner Diplomarbeit betreut und geholfen hat.

Ein weiterer Dank richtet sich an meine Teamkollegen, da unsere Zusammenarbeit in jedem Schritt des Projekts bestens funktioniert hat.

Zusätzlich will ich noch meiner Freundin danken, die mich bei der Überprüfung der Grammatik und Rechtschreibung unterstützt hat.

Zusammenfassung

Ziel der Diplomarbeit ist es, eine Anwendung zu entwickeln, die, mithilfe von künstlicher Intelligenz, Vorhersagen über die Entwicklung des DAX (= Deutscher Aktienindex) versteht und auswertet.

In dieser Diplomarbeit werden die Grundprinzipien des Webcrawlings vorgestellt. Dabei werden allgemeine Fragen, wie zum Beispiel „Was ist Crawling?“ oder „Wozu wird Webcrawling verwendet?“ beantwortet. Des Weiteren werden wichtige Themen, die eng im Zusammenhang mit dem Webcrawling stehen, wie beispielsweise Hyperlinks, das Crawl-Budget oder die Suchmaschinenoptimierung beschrieben.

Weiters wird technisch auf den Aufbau und den Arbeitsprozess eines Crawling-Programmes eingegangen. Zudem werden verschiedene Strategien, nach denen ein solches Programm verfahren kann und zusätzliche relevante Themen, wie der Robots Exclusion Standard oder Spider-Traps beschrieben. Frameworks zum Programmieren eines Crawlers und mögliche Anwendungsbereiche für Crawler werden ebenso vorgestellt.

Zuletzt wird die Implementation und Automatisierung eines Webcrawlers im Projekt AI Börse beschrieben.

Außerdem wird in der Diplomarbeit auch ein Überblick über die Verarbeitung von Big Data über Clustersysteme gegeben. Dazu werden die Grundlagen sowie die verschiedenen Arten von Clustern dargestellt. Es werden aber auch die Grenzen von modernen Clustern beschrieben.

Weiters werden Frameworks vorgestellt, welche mit Clustern arbeiten. Dabei liegt das Hauptaugenmerk auf den beiden Frameworks Apache Hadoop und Apache Spark, welche im Detail beschrieben werden. Es wird aber auch noch ein grober Eindruck von anderen Implementationen vermittelt.

Abschließend wird noch gezeigt, wie man Apache Spark MLlib Pipelines benutzen kann, um Aufgaben über eine AI zu bewältigen. Anschließend werden auch noch weitere Möglichkeiten gezeigt, um Apache Spark in das Projekt AI Börse zu implementieren.

Zudem wird auf die Computerlinguistik und ihre Geschichte eingegangen. Dabei werden allgemeine Prinzipien und Ziele der Computerlinguistik vorgestellt.

Des Weiteren wird dem Leser die Freitextanalyse mithilfe von Natural Language Processings nähergebracht. Dazu werden die Ebenen, Aufgaben und verschiedenen Ansätze von Natural Language Processing beschrieben.

Zu guter Letzt wird auf die verschiedenen Lösungsansätze eingegangen, mit denen im Projekt AI Börse versucht wurde, mithilfe von Natural Language Processing, automatisiert Prognosewerte aus Börsenbriefen auszulesen. Dabei macht ein Ansatz von Named-Entity-Recognition gebrauch, der zweite nutzt eine Sentiment-Analyse und der letzte verarbeitet die Texte prozedural.

Im letzten Abschnitt der Diplomarbeit wird erklärt was Daten sind und wozu diese gebraucht werden. Noch dazu werden dem Leser Begriffe wie Big Data, Data Analytics, Predictive Selling oder KI näher gebracht und mit Bildern oder Vergleichen erklärt.

Als Nächstes wird die Verarbeitung von Daten mittels des Begriffs CRUD erklärt. Um den Nutzen von Daten zu veranschaulichen dienen Beispiele im Bezug auf Börsen und Vergleiche mittels der Projektgruppe AI Börse. Nachdem dies verdeutlicht wurde, werden Unterschiede in der Struktur von Daten veranschaulicht und anschließend erklärt, wozu diese gebraucht und genutzt werden.

Zum Schluss des Abschnitts werden grafische Benutzeroberflächen (kurz genannt GUI) und deren Funktionen, Anforderungen und Bestandteile vorgestellt. Abschließend wird die Planung, Implementation und genauere Funktionen der GUI der Projektgruppe AI Börse mittels Codezeilen präsentiert.

Abstract

The goal of this thesis is to develop an application that, with the help of artificial intelligence, understands and evaluates predictions about the development of the DAX (= Deutscher Aktienindex (German stock index)).

In this thesis, the basic principles of web crawling are presented. Thereby, general questions, such as „What is crawling?“ or „What is web crawling used for?“ are answered. Furthermore, important topics closely related to web crawling are described, such as hyperlinks, the crawl budget or search engine optimization.

In addition, the structure and the working process of a crawling program are described. Various strategies according to which such a program can proceed and additional relevant topics such as the Robots Exclusion Standard or spider traps are described as well. Frameworks for programming a crawler and possible areas of application for crawlers are also presented.

Finally, the implementation and automation of a web crawler in the project AI Börse is described.

Furthermore, this thesis also gives an overview of the processing of Big Data via cluster systems. For this purpose, the basics as well as the different types of clusters are presented. However, the limitations of modern clusters are also described.

Moreover, frameworks that work with clusters are presented. The main focus is on the two frameworks Apache Hadoop and Apache Spark, which are described in detail. However, a rough impression of other implementations is also given.

Finally, it will be shown how to use Apache Spark MLlib pipelines to handle tasks via an AI. Afterwards, further possibilities are also shown to implement Apache Spark in the project AI Börse.

In addition, computational linguistics and its history are discussed. General principles and goals of computational linguistics are presented.

Furthermore, the reader is introduced to the analysis of natural text with the help of Natural Language Processing. The levels, tasks and different approaches of Natural Language Processing are described.

Finally, the various approaches used in the project AI Börse to automatically read forecast values from stock market letters with the help of Natural Language Processing will be discussed. One approach uses named entity recognition, the second uses sentiment analysis, and the last processes the texts procedurally.

The last section of the thesis explains what data is and what it is used for. In addition, the reader is introduced to terms such as Big Data, Data Analytics, Predictive Selling or AI and explained with images or comparisons.

Next, the processing of data is explained by means of the term CRUD. To illustrate the use of data, examples are used in relation to stock exchanges and comparisons using the project AI Börse. After this has been clarified, differences in the structure of data are illustrated and then it is explained what they are needed and used for.

At the end of the section, graphical user interfaces (GUI for short) and their functions, requirements and components are presented. Finally, the planning, implementation and more detailed functions of the GUI of the project AI Börse are presented by lines of code.

Inhaltsverzeichnis

Vorwort	i
Erklärung	i
Diplandenvorstellung	ii
Danksagungen	vi
Zusammenfassung	x
Abstract	xii
Inhaltsverzeichnis	xiv
1 Zielsetzung	1
1.1 Die Webüberwachungs- und Analyseapplikation AI Börse	1
1.2 Webcrawling - Nicolas Philipp	1
1.3 Clustersysteme - Louis Raschbach	2
1.4 Natural Language Processing - Simon Mader	2
1.5 Repräsentation von Daten - Alen Asanov	3
2 Allgemeines zu Webcrawling	4
2.1 Allgemeine Erklärung	4
2.2 Was bedeutet Crawling?	5

2.3	Was sind Hyperlinks?	5
2.4	Was bedeutet Indexierung?	10
2.5	Crawl-Budget & Index Budget	11
2.6	Suchmaschinenoptimierung	12
2.6.1	Onpage-Optimierung	13
2.6.2	Offpage-Optimierung	21
3	Crawler	23
3.1	Überblick	23
3.2	Anforderungen an einen Webcrawler	23
3.3	Komponenten eines Crawlers	25
3.4	Arbeitsprozess eines Crawlers	30
3.5	Crawlingstrategien	31
3.5.1	Breadth-First & Depth-First	31
3.5.2	Incremental Crawler	32
3.5.3	Focused Crawler	32
3.5.4	Distributed Crawler	33
3.6	Anwendungsbereiche für Crawler	34
3.7	Robots Exclusion Standard	36
3.7.1	Allgemein	36
3.7.2	Speicherort	37
3.7.3	Aufbau	37
3.8	Beispiele für Suchmaschinencrawler	42
3.8.1	Googlebot	42

3.8.2	Bingbot	42
3.8.3	DuckDuckBot	43
3.9	Spider-Traps	43
3.10	Webcrawling Frameworks	46
3.10.1	Scrapy	46
3.10.2	Jaunt	48
3.10.3	Goutte	49
4	Implementation eines Webcrawlers im Projekt AI Börse	51
4.1	Problemstellung	51
4.2	Spezifikation	52
4.3	Arbeitsprozess	53
4.4	Implementation	55
4.4.1	Linkcrawler	55
4.4.2	Contentcrawler	57
4.4.3	Libertexcrawler	59
4.5	Automatisierung	62
4.5.1	Crontab	62
4.6	Fazit	62
5	Clustersysteme	64
5.1	Was ist ein Clustersystem	64
5.2	Arten von Clustersystemen	65
5.3	Grenzen der einfachen Cluster	67
5.3.1	Amdahl's Law	67

6 Big Data Frameworks, die mit Clustersystem arbeiten	70
6.1 Apache Hadoop	70
6.1.1 Geschichte	71
6.1.2 Funktionsweise	72
6.1.3 Erweiterungen	77
6.1.4 Hadoop-Cluster	82
6.1.5 Fazit	82
6.2 Apache Spark	84
6.2.1 Geschichte	85
6.2.2 Verhältnis zu Hadoop	85
6.2.3 Funktionsweise	87
6.2.4 Pythonic	101
6.2.5 Fazit	103
6.3 Weitere Implementationen	106
6.3.1 Kubernetes	106
6.3.2 Mesos	107
6.4 Alternativen	107
6.4.1 Dremel	107
7 Implementation von Clustersystemen im Projekt AI Börse	109
7.1 Installation von PySpark auf einem Windows Server	109
7.2 Einbindung von PySpark in die Analyse von Börsenbriefe	115
7.2.1 Pipelines	115
7.2.2 Alternative Einbindungen von Clustersystemen	120

8 Computerlinguistik	124
8.1 Allgemeines	124
8.1.1 Ziele der Computerlinguistik	124
8.2 Geschichte	125
8.2.1 1950 bis 1969	125
8.2.2 1970 bis 1989	126
8.2.3 Ab 1990	126
9 Natural Language Processing	128
9.1 Allgemeines	128
9.2 Ebenen von Natural Language Processing	129
9.2.1 Syntax	129
9.2.2 Morphologie	131
9.2.3 Semantik	131
9.2.4 Pragmatik	133
9.2.5 Diskurs	133
9.2.6 Phonologie	134
9.3 Aufgaben von Natural Language Processing	135
9.3.1 Tokenisierung	135
9.3.2 Named Entity Recognition	135
9.3.3 Part-of-Speech-Tagging	136
9.3.4 Text-Klassifikation	137
9.3.5 Textgenerierung	138
9.3.6 Question Answering	139

9.3.7	Maschinelle Übersetzung	140
9.4	Symbolisches Natural Language Processing	141
9.4.1	Allgemeines	141
9.5	Statistisches Natural Language Processing	142
9.5.1	Allgemeines	142
9.5.2	Machine Learning	144
9.6	Natural Language Processing und künstliche neuronale Netze	160
9.6.1	Was ist ein künstliches neuronales Netz?	160
9.6.2	Arten von künstlichen neuronalen Netzen	163
10	Analyse von Börsennews im Projekt AI Börse	168
10.1	Aufgabenstellung	168
10.2	Analyse mithilfe von Named-Entity-Recognition	170
10.2.1	Verwendete Technologien	170
10.2.2	Implementation	171
10.2.3	Problem	174
10.3	Sentiment-Analyse der Börsenbriefe	174
10.3.1	Verwendete Technologien	174
10.3.2	Implementation	175
10.3.3	Problem	180
10.4	Analyse mithilfe eines prozeduralen Ansatzes	180
10.4.1	Verwendete Technologien	181
10.4.2	Implementation	181
10.4.3	Fazit	184

11 Allgemeines über Daten	185
11.1 Definition	185
11.2 Wozu braucht man Daten?	186
11.3 Big Data	188
11.3.1 Data Analytics	190
11.3.2 Predictive Selling	192
11.4 Künstliche Intelligenz	193
11.4.1 Definition	193
11.4.2 Die Geschichte hinter der künstlichen Intelligenz	193
11.4.3 Arten von KI	194
11.5 Verarbeitung von Daten	196
11.5.1 Was ist CRUD?	196
11.5.2 Unterscheidung der Daten	198
11.5.3 Interpretation und Analyse von Kursverläufen	199
12 Repräsentation von unformatierten Daten	205
12.1 Strukturen von Daten	205
12.1.1 Was sind unformatierte Daten?	205
12.1.2 Struktur der Daten	205
12.2 Darstellung von Daten	212
12.2.1 Wieso werden Daten dargestellt?	212
12.3 Grafische Benutzeroberfläche	217
12.3.1 Definition einer GUI	217
12.3.2 Funktion einer GUI	217

12.3.3	Design und Qualität einer GUI	221
12.4	Vergleich mit dem Projekt AI Börse	227
12.4.1	Planung	227
12.4.2	Codelgniter	229
12.4.3	Was ist Codelgniter?	229
12.4.4	Implementation der GUI	231
12.4.5	Programmierung der GUI	235
12.4.6	Fazit	244
Anhang		245
	Abbildungsverzeichnis	245
	Verzeichnis der Listings	249
	Index	253
	Literaturverzeichnis	253
	Kapitelzuordnung	272
	Begleitprotokolle	273
	Besprechungsprotokolle	283

Kapitel 1

Zielsetzung

1.1 Die Webüberwachungs- und Analyseapplikation AI Börse

Es soll eine Anwendung entwickelt werden, die mit Hilfe von künstlicher Intelligenz, Vorhersagen über die Entwicklung des DAX versteht, und auswertet. Das Ziel der Anwendung ist, eine Schätzung, aufgrund vorheriger Prognosen, zu liefern, welche Börsennewsseite am ehesten die Entwicklung des DAX vorhersagt.

Mit einem Crawler werden täglich Inhalte und Texte von Webseiten, welche Prognosen zur Entwicklung des DAX veröffentlichen, ausgelesen. Der Freitext wird anschließend mit NLP (= Natural Language Processing) formatiert und aufbereitet, sodass eine AI (= Artificial Intelligence, künstliche Intelligenz) diesen vorbereiteten Text verarbeiten und verstehen kann.

Die Ergebnisse der AI, und die tatsächlichen DAX-Kurs Ergebnisse werden in einer Datenbank abgespeichert und anschließend miteinander verglichen. In einer GUI (= Graphical User Interface) wird dieser Vergleich dargestellt, und in Tabellenform wird angezeigt, welche Quelle in einem ausgewählten Zeitraum (7 Tage, 30 Tage, 90 Tage, ...) die besten Prognosen geliefert hat.

1.2 Webcrawling - Nicolas Philipp

Ziel dieses Teils der Arbeit ist es, ein gewisses Grundwissen, was das Webcrawling betrifft, aufzubauen. Es sollen allgemeine Themen wie Hyperlinks, das Crawl-Budget

und die Suchmaschinenoptimierung für das bessere Verständnis beschrieben werden. Weiters soll die Architektur eines Webcrawlers, dessen Anwendungsbereiche und dessen Arbeitsprozess aufgezeigt werden. Zudem sollen der Robots Exclusion Standard, Crawling-Strategien und Spider-Traps behandelt werden.

Die praktische Umsetzung des Crawlers soll mit einem Webcrawling-Framework, welches ebenfalls in der Arbeit beschrieben wird, erfolgen. Die Ergebnisse der Implementation sollen außerdem ausführlich dargestellt werden.

1.3 Clustersysteme - Louis Raschbach

Ziel dieses Abschnittes ist es, einen groben Überblick über das Thema Clustersysteme zu geben. Wobei die Grundstrukturen, Grenzen und verschiedenen Arten von Clustern erklärt werden.

Weiteres sollen noch die beiden großen Frameworks, welche mit Clustersystemen arbeiten, Apache Hadoop und Apache Spark beschrieben werden. Dabei soll erklärt werden, wie diese beiden Frameworks arbeiten und welche Funktionen sie mit sich bringen.

Abgeschlossen wird dieses Kapitel mit der praktischen Implementierung von Clustersystem über Apache Spark, durch die Verwendung von Pipelines und PySpark DataFrames.

1.4 Natural Language Processing - Simon Mader

Ziel dieses Kapitels ist es, einen allgemeinen Überblick über die Computerlinguistik und der Freitextanalyse mithilfe von Natural Language Processing zu schaffen. Dabei werden, zusätzlich zu den Grundprinzipien und Zielen der Computerlinguistik, die Ebenen, Aufgaben und Ansätze im Bereich Natural Language Processing erklärt.

Bei der praktischen Umsetzung eines Programms, das Natural Language Processing nutzt, um automatisch aus Börsenbriefen Prognosewerte auszulesen, wird auf drei verschiedene Ansätze eingegangen, mit denen versucht wurde diese Funktionalität umzusetzen. Dabei wird bei jedem Ansatz auf die verwendeten Technologien, die Implementation und, gegebenenfalls, auf Probleme eingegangen.

1.5 Repräsentation von Daten - Alen Asanov

In diesem Teil der Arbeit soll dem Leser erklärt werden, was unformatierte Daten sind und was der Unterschied zu normalen (formatierten) Daten ist. Um dies zu erreichen wird mit dem Erklären des allgemeinen Begriffs Daten begonnen und anschließend mit dem Beschreiben des Nutzens fortgefahren. Große Begriffe wie Big Data und KI sollen dem Leser dabei näher gebracht werden, indem man sie mittels Vergleiche und Beispiele erklärt.

Nachdem das Wissen über unformatierte Daten vermittelt wurde, kann veranschaulicht werden, wie Daten dargestellt werden können und worauf hierbei geachtet werden sollte. Dabei wird die GUI angesprochen und ihre Funktionen und Elemente näher gebracht.

Am Schluss der Arbeit wird der praktische Teil mittels des CodeIgniter-Frameworks beschrieben. In der Dokumentation der praktischen Arbeit soll die Planung, Implementation und Codebeschreibung erklärt werden.

Kapitel 2

Allgemeines zu Webcrawling

2.1 Allgemeine Erklärung

Damit Webseiten in den Suchergebnissen einer Suchmaschine angezeigt werden können, benötigt es Programme, die die relevanten Inhalte einer Webseite automatisiert durchsuchen und in den Index aufnehmen. Solch eine Anwendung wird Webcrawler, oft aber auch Spider, Searchbot, Robot, oder nur Crawler genannt. Eine flache Seitenhierarchie und eine durchdachte interne Verlinkung per Hyperlinks begünstigen die gewollte Aufnahme der Webseite in den Index der Suchmaschine. (NP:Web01, vgl. more-fire.com 27.10.2020)

Der Index ist ein riesiger Datenbestand an Milliarden von gecrawlten Webseiten, welcher bei einer Suchanfrage des Benutzers geeignete Suchergebnisse liefert. Webcrawler dienen also dazu, dass Inhalte auf Webseiten durch die User der Suchmaschinen gefunden werden. (NP:Web01, vgl. more-fire.com 27.10.2020)

Crawler finden aber auch anderweitig Einsatz, beispielsweise in der automatisierten Datenbeschaffung → hierbei werden die Programme darauf spezialisiert, kontinuierlich Daten von Webseiten auszulesen, um diese dann in gewünschtem Format bereitzustellen. Mehr dazu in Abschnitt 3.6.

In den folgenden Kapiteln werden Begriffe rund um das Crawling erklärt, um ein Vorwissen aufzubauen, welches für das bessere Verständnis des eigentlichen Themas, das Webcrawling, benötigt wird.

2.2 Was bedeutet Crawling?

Das Crawling bezeichnet den Arbeitsprozess und jeden damit verbundenen Vorgang eines Crawlers. Ein Suchmaschinencrawler folgt dabei allen auf den Webseiten auffindbaren Verlinkungen (Hyperlinks), um so möglichst viele Seiten zu indexieren. Sprich, diese für die Öffentlichkeit zugänglich zu machen. Näheres zur Arbeitsweise eines Crawlers folgt im Abschnitt 3.4.

2.3 Was sind Hyperlinks?

Grundsätzlich sind Hyperlinks, kurz Links, Verknüpfungen, die von einer Webseite auf eine andere Seite verweisen oder innerhalb einer Seite bestimmte Funktionen auslösen. Hyperlinks werden standardmäßig blau hinterlegt und unterstrichen dargestellt. (NP:Web06, vgl. html-seminar.de 30.10.2020)

Um Links innerhalb einer Webseite einzubinden, müssen sie auf folgende Weise im HTML¹-Body implementiert werden:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>Beispiel</title>
5      </head>
6      <body>
7          <a href="https://www.beispiel.at">Beispieltext</a>
8      </body>
9  </html>
```

Listing 2.1: Hyperlink auf einer Webseite einbinden

Der Text, der zwischen Start- und End-Tag des Anchor-Elements (a-Tag) steht, wird als klickbares Textelement dargestellt. Mit dem “href”²-Attribute wird angegeben, auf welche Seite das Element verweisen soll; beziehungsweise welche Funktion das Textelement haben soll → es stellt somit den Verweis dar.

Hyperlinks müssen nicht zwangsweise als Textelement dargestellt werden, man kann alternativ auch ein Bild mit solch einem Link hinterlegen; dazu wird statt dem Text der angezeigt werden soll, einfach mittels img-Tag das Bild eingefügt. (NP:Web09, vgl. ionos.at 05.11.2020)

¹Hypertext Markup Language

²hyper reference

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Beispiel</title>
5   </head>
6   <body>
7     <a href="https://www.beispiel.at"></a>
8   </body>
9 </html>
```

Listing 2.2: Bild mit Hyperlink hinterlegen

Interne Links

Interne Links sind Verweise auf Unterseiten innerhalb einer Webseite. (NP:Web08, vgl. [advidera.com](#) 30.10.2020)

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Beispiel</title>
5   </head>
6   <body>
7     <a href="information.html">Informationen</a>
8   </body>
9 </html>
```

Listing 2.3: Verweis auf eine Unterseite

In obigem Beispiel wird daher mit einem Klick auf das Textelement “Informationen” von der aktuellen Seite auf die Unterseite “information.html” gesprungen.

Externe Links

Bei externen Links wird zwischen zwei Arten unterschieden:

- **Outbound Links**

Outbound Links sind die Links, die auf Webseiten verweisen, die nicht zu der eigenen Domain (Internetadresse) gehören; sie sind ausgehende externe Links. (NP:Web10, vgl. [eology.de](#) 05.11.2020)

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>Beispiel</title>
5      </head>
6      <body>
7          <a href="https://www.htlstp.ac.at/">HTL St. Pölten</a>
8      </body>
9  </html>

```

Listing 2.4: Outbound Link im HTML-Body

Der Hyperlink aus obigem Beispiel führt demnach auf die Webseite der HTL St. Pölten.

- **Backlinks**

Die zweite Art der externen Links sind Backlinks; sie sind eingehende externe Links. Von so einem Link spricht man, wenn von einer externen Domain auf die eigene Domain verwiesen wird. Sprich Outbound- und Backlink sind an sich gleich, nur die Betrachtungsweise unterscheidet sie. (NP:Web10, vgl. eology.de 05.11.2020)

Links mit anderen Funktionen

Hyperlinks können auch andere Funktionen als nur das Verweisen auf eine interne oder externe Seite haben. Weitere Anwendungen sind etwa das Öffnen eines PDFs, oder das Öffnen eines E-Mail-Programmes. (NP:Web08, vgl. advidera.com 30.10.2020)

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>Beispiel</title>
5      </head>
6      <body>
7          <a href="https://www.beispiel.at/beispiel.pdf#page=xx">PDF</a>
8      </body>
9  </html>

```

Listing 2.5: Hyperlink um ein PDF zu öffnen

Um ein PDF per Hyperlink zu öffnen, gibt man als "href"-Attribut den Speicherort der gewünschten Datei an. Optional kann man mit dem Zusatz "page=" festlegen, bei welcher Seite innerhalb des Dokuments gestartet werden soll.

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Beispiel</title>
5   </head>
6   <body>
7     <a href="mailto:n.philipp@htlstp.at">Email schreiben</a>
8   </body>
9 </html>

```

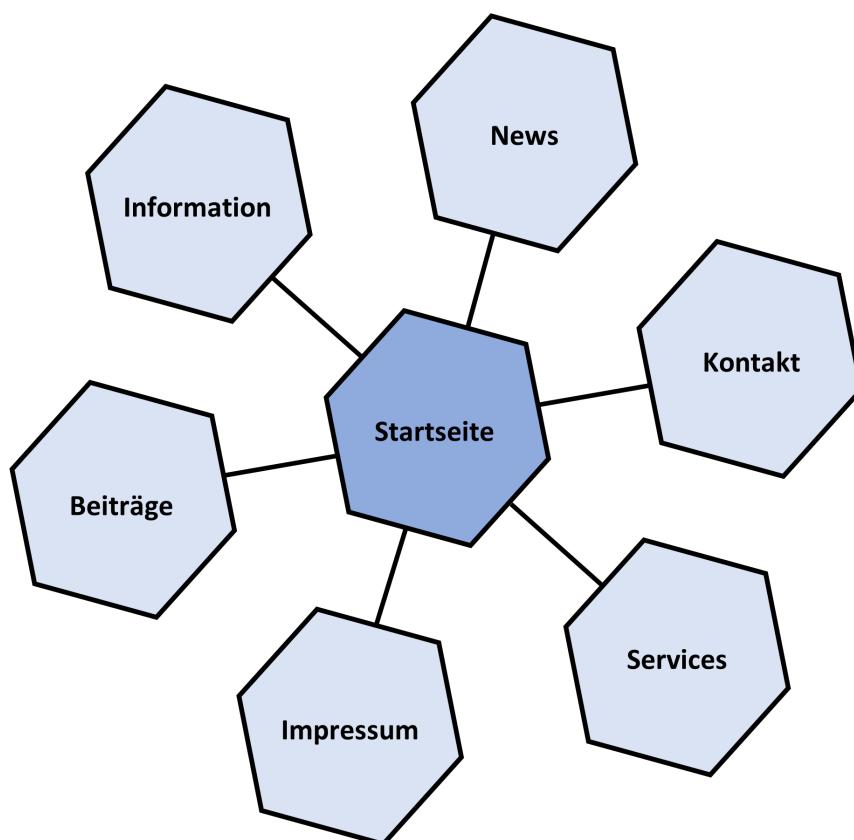
Listing 2.6: Hyperlink zum Öffnen eines E-Mail-Programmes

In obigem Code-Beispiel wird gezeigt, wie der Verweis zum Öffnen eines E-Mail-Programmes funktioniert. Das "href"-Attribut bekommt hierbei den Wert "mailto:e-mail". Die E-Mail-Adresse welche hier eingetragen wird, ist die Adresse die dann automatisch im E-Mail-Programm als Empfänger drinsteht.

Hyperlinkstrukturen

- **Sternstruktur**

Mithilfe der Sternstruktur werden verschiedene themenrelevante Verlinkungen auf einer Seite gemacht. Diese Verweise sind nicht unbedingt Links auf Seiten in der eigenen Domain, sondern können auch extern sein. Die Struktur kommt meist in digitalen Lexika zum Einsatz. (NP:Web10, vgl. eology.de 05.11.2020)

Abbildung 2.1: Sternstruktur
Angelehnt an: (NP:Web10, vgl. eology.de 05.11.2020)

- **Lineare Struktur**

Bei der linearen Struktur wird der Benutzer mit einem Klick auf einen Hyperlink auf eine bestimmte Webseite weitergeleitet → der User kann dabei nicht beeinflussen wo er landet, da die Reihenfolge vom Webseitenbetreiber festgelegt ist. (NP:Web10, vgl. eology.de 05.11.2020)

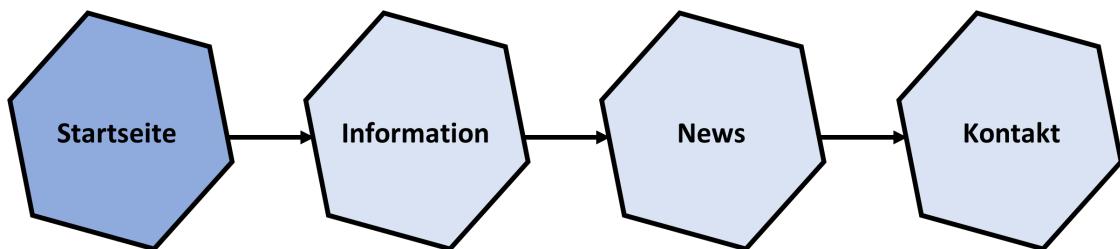


Abbildung 2.2: Lineare Hyperlinkstruktur
Angelehnt an: (NP:Web10, vgl. eology.de 05.11.2020)

- **Netzstruktur**

Diese Struktur ermöglicht es dem Benutzer von jeder Seite aus auch alle anderen Seiten zu besuchen. Solch eine Struktur findet man meist bei Online-Shops da dort auf jeder Unterseite auch das Menü angezeigt wird. (NP:Web10, vgl. eology.de 05.11.2020)

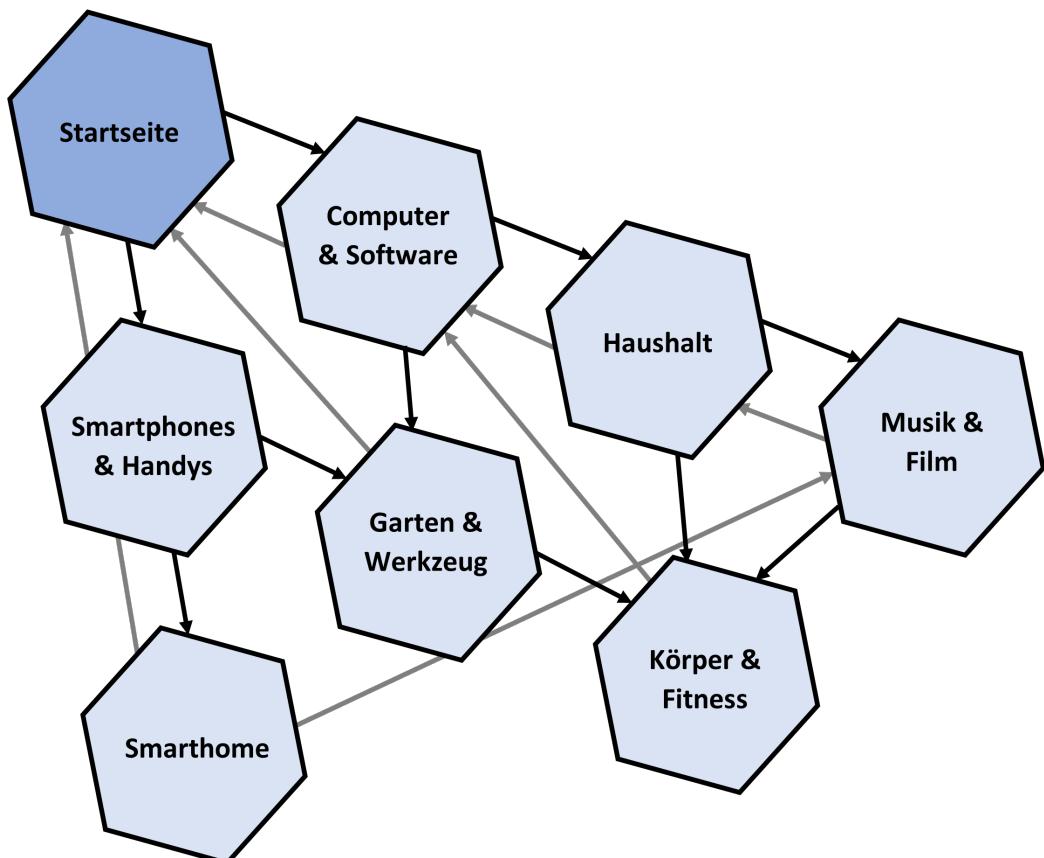


Abbildung 2.3: Netzstruktur
Angelehnt an: (NP:Web10, vgl. eology.de 05.11.2020)

- **Baumstruktur**

Der Benutzer kommt von einer Oberseite auf die weiteren Unterseiten. Die Baumstruktur ist typisch für Webseiten bei denen man mittels einer Navigationsleiste bestimmte Inhalte der Seite abfragen kann. (NP:Web10, vgl. eology.de 05.11.2020)

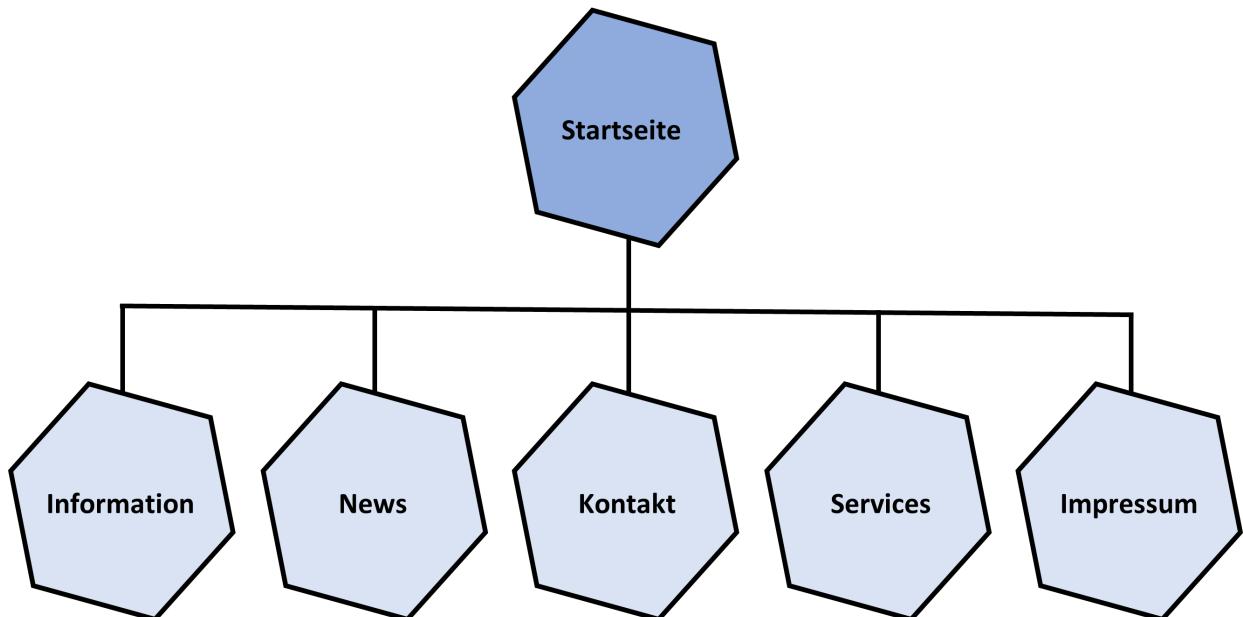


Abbildung 2.4: Baumstruktur
Angelehnt an: (NP:Web10, vgl. eology.de 05.11.2020)

2.4 Was bedeutet Indexierung?

Die Indexierung im Zusammenhang mit Suchmaschinen bezeichnet die Sammlung von Daten, denen auf Basis der Inhalte bestimmte Schlagwörter zugeordnet werden. Diese Sammlung bildet den Index, der dafür sorgt, dass dem Benutzer einer Suchmaschine relevante Ergebnisse bei der Suche angezeigt werden. Suchmaschinen wie zum Beispiel Google, Bing oder Yahoo suchen demnach bei einer Anfrage des Users nicht das gesamte Internet ab, sondern nur ihren Index. (NP:Web02, vgl. xovi.de 28.10.2020)

Da jede Suchmaschine ihren eigenen Index besitzt, bekommt man auch je nachdem, welche man verwendet, unterschiedliche Suchergebnisse. Die Aktualisierung eines solchen Index erfolgt meist kontinuierlich, da so neue Inhalte rasch aufgenommen werden können und die Qualität der Suchergebnisse verbessert werden kann. (NP:Web02, vgl. xovi.de 28.10.2020)

2.5 Crawl-Budget & Index Budget

Vereinfacht gesagt handelt es sich beim Crawl-Budget um die Anzahl der URLs, die der Bot auf einer Seite crawlten kann und crawlten will. Dieses Budget setzt sich aus der Crawling-Frequenz (Können) und dem Crawling-Bedarf (Wollen) zusammen. (NP:Web01, more-fire.com 27.10.2020)

Die Anzahl der Anfragen pro Sekunde, die der Crawler während des Crawlingvorganges auf der Seite ausführt, nennt man in diesem Zusammenhang Frequenz. Diese wird durch niedrige Fehleranfälligkeit und kurze Ladezeiten der Webseite positiv beeinflusst. (NP:Web01, vgl. more-fire.com 27.10.2020)

Unter dem Crawling-Bedarf versteht man eine Kennzahl, die aussagt, ob eine Webseite regelmäßig gecrawlt werden soll. Duplicate Content, Soft-404-Fehler oder Spam führen dazu, dass die Seite als unwichtig eingestuft wird. (NP:Web01, vgl. more-fire.com 27.10.2020)

Weiters wird das Crawl-Budget auch großteils durch den PageRank einer Webseite festgelegt. Der PageRank ist eine Bewertung einer Seite auf Basis ihrer eingehenden Verlinkungen. Dabei wird die Seite besser bewertet, unabhängig vom Inhalt, wenn viele und gleichzeitig auch "wichtige" Seiten, also Seiten mit einem hohen PageRank, auf diese verlinken. Dementsprechend wird der PageRank rekursiv mithilfe den Werten der vorhergehenden Seiten berechnet. Je höher der PageRank ist, umso größer ist auch das Budget, sprich es werden mehr Kapazitäten für das Indexieren dieser Seite verwendet. (NP:Web03, NP:Web04, vgl. de.ryte.com 28.10.2020)

Die Formel für den PageRank einer Seite X lautet also wie folgt:

$$PR(X) = (1 - d) + d \cdot \left(\frac{PR(T1)}{C(T1)} + \dots + \frac{PR(Tn)}{C(Tn)} \right)$$

PR(X) ... PageRank der Seite X

d ... Dämpfungsfaktor, welcher alle Werte zwischen 0 und 1 annehmen kann

PR(T1) bis PR(Tn) ... PageRank der auf X verlinkenden Seiten T1 bis Tn

C(T1) bis C(Tn) ... Anzahl der Links auf der jeweiligen Seite T1 bis Tn

Die Berechnung erfolgt nach dem Random Surfer Model: Die Wahrscheinlichkeit, dass ein Nutzer eine bestimmte Seite besucht, hängt von der Anzahl der Links, die auf genau diese Seite verweisen, ab. Die Wahrscheinlichkeit, dass der User diese bestimmte Webseite besucht, ist also die Summe der Wahrscheinlichkeiten, mit der der Surfer auf die Seite durch ihre Verlinkungen stößt. (NP:Web05, vgl. de.ryte.com 28.10.2020)

Dadurch, dass ein Nutzer nicht unendlich Links folgt, wird der Wert um den Dämpfungs-faktor d verringert. Dieser Faktor stellt dabei die Wahrscheinlichkeit dar, dass der User einem Link folgt. Je höher der Wert, desto eher wird einem Link gefolgt. Das $(1-d)$ aus der Formel berechnet die Wahrscheinlichkeit, dass der Benutzer eine beliebige neue Seite aufruft. (NP:Web05, vgl. de.ryte.com 28.10.2020)

Der PageRank-Algorithmus verliert heutzutage zunehmend an Bedeutung, da mittler-weile mehr Faktoren bei der Bewertung einer Webseite Einfluss haben, wie zum Beispiel die durchschnittliche Aufenthaltsdauer³ der User oder die Absprungrate (Bounce Rate, → wie viel Prozent der Benutzer nur eine Seite auf der Domain aufrufen und diese dann wieder verlassen). Die Webseiten, die als unbeliebt, veraltet oder unwichtig eingestuft werden, werden selten bis gar nicht gecrawlt und somit nicht in den Index aufgenommen, was wiederum bedeutet, dass sie über die Suchmaschine nicht gefunden werden. (NP:Web04, vgl. de.ryte.com 28.10.2020)

Unterschied zum Index-Budget

Beim Index-Budget handelt es sich um einen Kennwert, welcher festlegt, wie viele URLs von der Suchmaschine indexiert werden können. Der Unterschied zum Crawl-Budget macht sich dadurch erkennbar, dass eine Seite, welche einen Fehlercode zurückgibt, nicht in den Index aufgenommen wird und somit das Index-Budget nicht belastet. Hingegen werden Ressourcen aus dem Crawl-Budget verwendet, da die jeweilige Seite aufgerufen wird. (NP:Web03, vgl. de.ryte.com 28.10.2020)

2.6 Suchmaschinenoptimierung

Jeder Webseitenbesitzer möchte, dass seine Webseite in den Suchergebnissen ganz oben steht, sei es jetzt ein Online-Shop oder ein einfacher Blog, doch wie kann beeinflusst werden, wie gut die Webseite in den Suchergebnissen platziert (geranked) wird?

Mithilfe von Suchmaschinenoptimierung können Webseitenbetreiber selbst Best Practices anwenden, um damit ihr Ranking eventuell zu verbessern. Die Optimierung ist dabei ein kontinuierlicher Verbesserungsprozess, in welchem die Webseite laufend analysiert und angepasst werden muss. Während des Prozesses wird jedenfalls zwischen On- und Offpage-Optimierung unterschieden. (NP:Web11, vgl. de.ryte.com 09.11.2020)

³Average Time on Site

2.6.1 Onpage-Optimierung

Die Onpage-Optimierung bezieht sich auf die Optimierung von zentralen technischen, inhaltlichen und strukturellen Aspekten der Webseite (onpage → auf der Seite). Das Ranking soll durch einfach crawlaren und indexierbaren Content verbessert werden. Auch die allgemeine User Experience⁴ profitiert von guter Onpage-Optimierung. (NP:Web12, vgl. de.ryte.com 09.11.2020)

In den folgenden Abschnitten werden die einzelnen Bereiche der Onpage-Optimierung erläutert.

Webseiten- und URL-Struktur

Je öfter der Benutzer auf einer Webseite weiterklicken muss, um von der Startseite zum betreffenden Content zu gelangen, desto größer wird auch die Klicktiefe. Wenn die Webseite eine zu tiefe Untergliederung hat und dementsprechend auch eine große Klicktiefe hat, werden die Inhalte möglicherweise nur unregelmäßig vom Crawler gefunden und Änderungen werden somit nicht wünschenswert indexiert. (NP:Web13, vgl. vioma.de 09.11.2020)

Die Adresse der Webseite (URL, Uniform Resource Locator) muss auch einigermaßen optimiert werden. URLs sollten möglichst kurz gehalten und “sprechend” gestaltet werden. Kurze Adressen werden erfahrungsgemäß öfter angeklickt und von Benutzern geteilt. (NP:Web12, NP:Web13, vgl. de.ryte.com, vioma.de 09.11.2020)

Eine “sprechende” URL gibt auch schon beim Lesen der Zeile Auskunft darüber, was sich hinter dieser befindet. Meist kommt das Haupt-Keyword, ein bestimmtes Schlagwort, welches dieser Seite zuordenbar ist, schon in der URL vor, sodass diese relativ aussagekräftig ist. (NP:Web12, NP:Web13, vgl. de.ryte.com, vioma.de 09.11.2020)

Eine URL der Form:

https://www.beispiel.at/9237-pageID_D3j90aEMe1/artikel14.html

wird beispielsweise optimiert zu:

<https://www.beispiel.at/Onpage-Optimierung/URL-Struktur/>

⁴Nutzererfahrung

HTML-Code

Title-Tag

Im Title-Tag wird der allgemeine Titel der Webseite eingetragen. Es ist eines der wichtigsten Meta-Elemente einer Seite und wird in den Suchergebnissen immer als Überschrift für die Webseite verwendet. Der Title-Tag spielt eine große Rolle wenn es um das Ranking der Seite geht, deswegen muss dieser ein zentrales Keyword oder eine entsprechende Kombination aus Keywords der Zielseite beinhalten. (NP:Web12, vgl. de.ryte.com 09.11.2020)



Abbildung 2.5: Title-Tag des Ryte Wiki

Meta-Description-Tag

In die Meta-Description wird eine Zusammenfassung, welche die jeweilige Unterseite der Webseite beschreiben soll, hineingeschrieben. Dabei hat diese Bewertung keinen Einfluss auf die Bewertung durch die Suchmaschine, jedoch kann durch aussagekräftige Beschreibungen das Interesse des Users geweckt werden und daraus resultierend mehr Traffic⁵ auf der Seite verursacht werden. Auch hier gilt wieder → wichtige Keywords verwenden und diese hervorheben, beispielsweise durch fetten Text. (NP:Web12, vgl. de.ryte.com 09.11.2020)



Abbildung 2.6: Description-Tag des Ryte Wiki

⁵Netzwerkverkehr

Canonical-Tag

Wenn man denselben oder fast denselben Content auf mehreren Seiten innerhalb der Domäne anzeigen lässt, kommt es beim Crawling durch die Suchmaschine zu Duplicate Content. Die Suchmaschine kann jedoch nicht einfach entscheiden, welche der Unterseiten geranked und in den Suchergebnissen erscheinen soll. Dies kann zur Folge haben, dass eine Webseite eventuell mit mehreren URLs zu einem Keyword geranked wird, aber mit keiner dieser Seiten relativ gut. Eine weitere Folge kann aber auch sein, dass eine Seite für ein Schlagwort geranked wird, wofür sie eigentlich gar nicht vorgesehen war. Mit einem Canonical-Tag kann man der Suchmaschine zeigen, welche der URLs zur Indexierung verwendet werden soll. (NP:Web14, NP:Web15, vgl. de.ryte.com, developers.google.com 13.11.2020)

Anwendungsfälle für das Canonical-Tag:

- Startseite ist über verschiedene URLs abrufbar (z.B. www.beispiel.at, Beispiel.at, www.beispiel.at/index.html, ...)
- Seiten mit und ohne Trailing Slashes (“/”) abrufbar
- Seiten mit Klein- und Großschreibung abrufbar
- Unterschiedliche URLs, die sich durch Rewriting der URL durch den Server ergeben
- Inhalt der Seiten in verschiedenen Formen abrufbar (z.B. PDF, ...)
- Content wird extern auf anderen Seiten ebenfalls veröffentlicht
- ...

(NP:Web14, vgl. de.ryte.com 13.11.2020)

Grundsätzlich gibt es zwei verschiedene Möglichkeiten eine Canonical-URL zu erzeugen. Bei ersterer Variante wird ein “link”-Element im HMTL-Head der jeweiligen Seite, die nicht indexiert werden soll (die Seite, auf der der Duplicate Content auftritt), eingetragen (NP:Web14, NP:Web15, vgl. de.ryte.com, developers.google.com 13.11.2020):

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>Beispiel</title>
5          <link rel="canonical" href="http://www.beispiel.at/seite.html">
6      </head>
7      <body>
8          Beispielcontent
9      </body>
10     </html>
```

Listing 2.7: Canonical-URL mit Link-Tag erzeugen

Das “rel”-Attribut mit dem Wert “canonical” gibt an, dass die URL, die als Wert für das “href”-Attribut eingetragen ist, eine Canonical-URL ist und auf den Inhalt verweist, der zur Indexierung verwendet werden soll.

Die zweite Variante wird angewandt, wenn die Standardressource eine PDF-Datei oder ein anderer Dateityp ist. Die Implementation erfolgt nicht über das “.html”-File der Seite, sondern ein “Link”-Eintrag wird in den HTTP⁶-Header der vom Server bei einer Anfrage vom Client (Browser, Suchmaschine) zurückgesendet wird, eingebunden (NP:Web14, NP:Web15, vgl. de.ryte.com, developers.google.com 13.11.2020):

```

1  HTTP/... 200 OK
2  Content-Type: application/pdf
3  ...
4  Link: <http://www.beispiel.at/beispiel.pdf>; rel="canonical"
5  ...
```

Listing 2.8: Erzeugen einer Canonical-URL per HTTP-Header

Bei beiden Varianten gilt es zu beachten, dass absolute Pfade und keine relativen Pfade verwendet werden dürfen.

Beispiel:

Absoluter Pfad: <http://www.beispiel.at/beispiel.pdf>

Relativer Pfad: /beispiel.pdf

⁶Hypertext Transfer Protokoll

Heading-Tag

Das Verwenden von Heading-Tags⁷ im HTML-Body erleichtert es der Suchmaschine, die Struktur der Seite zu erkennen und somit den Inhalt dieser besser zu erfassen. Pro Seite soll nur ein h1-Tag verwendet werden, welcher das Keyword beinhaltet. Andere h-Tags (h2 bis h6) werden mit absteigender Wichtigkeit verwendet. (NP:Web12, vgl. de.ryte.com 09.11.2020)

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <title>Beispiel</title>
5    </head>
6    <body>
7      <h1>Beispiel</h1>
8      <h2>Beispiel</h2>
9      <h3>Beispiel</h3>
10     <h4>Beispiel</h4>
11     <h5>Beispiel</h5>
12     <h6>Beispiel</h6>
13   </body>
14 </html>
```

Listing 2.9: Heading-Tags im HTML-Body



Abbildung 2.7: Heading-Tags h1 - h6

⁷Überschriften in HTML

ALT-Attribut

Wenn der Browser bestimmte Medien wie Bilder oder Videos nicht darstellen kann, dann wird der "ALT"-Text für dieses Element angezeigt. Dieser Text soll den Inhalt des Mediums repräsentieren. Für Suchmaschinencrawler ist dies von Vorteil, da ihnen dadurch der Inhalt des Elements beschrieben wird und es so auch möglich ist, beispielsweise ein Bild in der Bildersuche ranken zu lassen. (NP:Web12, vgl. de.ryte.com 09.11.2020)

Hervorhebung der Keywords

Wichtige Schlagwörter, zu denen die Webseite im Internet gefunden werden sollen, sollten im Text fett gedruckt oder in kursiv geschrieben werden. Neben dem Aspekt, dass es dem Content der Seite zusätzlich Struktur verleiht, gibt es der Suchmaschine auch wichtige Impulse zum Erfassen des Inhalts. (NP:Web12, vgl. de.ryte.com 09.11.2020)

Crawling und Indexierung der Seite

robots.txt

Mithilfe der "robots.txt"-Datei, die im Root-Verzeichnis liegen sollte, kann man beeinflussen, welche Bereiche der Domain vom Crawler besucht werden dürfen. Nur durch diese Datei selbst kann man den Zugriff für die Crawler nicht sperren, dazu benötigt es weitere Sicherheitsmaßnahmen. Außerdem beachten Crawler, die etwa für das E-Mail-Harvesting⁸ verwendet werden, diese Vorgaben schon gar nicht. Näheres zu diesem Thema bezüglich dem Aufbau der Datei und weiteren Informationen im Abschnitt 3.7. (NP:Web12, vgl. de.ryte.com 09.11.2020)

Sitemap.xml

Alle URLs, die indexiert werden sollen, können mithilfe der "Sitemap.xml"-Datei in der Google Search Console⁹ gespeichert werden. Damit signalisiert man dem Google Bot¹⁰, dass diese bestimmten URLs gecrawlt werden sollen. Die Chance, dass diese dann im Endeffekt auch gecrawlt werden, erhöht sich dadurch deutlich. Auch wenn die Inhalte überarbeitet wurden oder neue Unterseiten veröffentlicht werden sollen, ist es empfehlenswert, diese der "Sitemap.xml"-Datei hinzuzufügen, da sie auf diese Weise für die Nutzer schneller öffentlich zugänglich werden. (NP:Web12, vgl. de.ryte.com 09.11.2020)

⁸ Automatisches Auslesen von E-Mails im Internet

⁹ Analyse- und Servicetool für Webseiten

¹⁰ Crawler, der für die Indexierung für Google zuständig ist

Beispielaufbau einer XML-Sitemap (NP:Web27, sitemaps.org 07.01.2021):

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
3   <url>
4     <loc>http://www.example.com/</loc>
5     <lastmod>2005-01-01</lastmod>
6     <changefreq>monthly</changefreq>
7     <priority>0.8</priority>
8   </url>
9 </urlset>
```

Listing 2.10: Beispielaufbau einer XML-Sitemap

Die Sitemap wird mit dem urlset-Tag geöffnet und geschlossen, außerdem wird innerhalb dieses Tags der Namespace festgelegt. Mit einem url-Tag werden innerhalb des URL-Sets einzelne URL-Einträge für die Seiten, die gecrawlt werden sollen, erstellt. Innerhalb des url-Tags werden einzelne Parameter zur URL angegeben. Das loc-Tag steht dabei für die eigentliche URL. Es ist wichtig, dass der Wert mit dem jeweiligen Protokoll (z.B.: HTTP, HTTPS, ...) beginnt, mit einem Schrägstrich endet und maximal 2048 Zeichen lang ist. (NP:Web27, vgl. sitemaps.org 07.01.2021)

Weiters können die Parameter "lastmod", "changefreq" und "priority" angegeben werden. Ersterer steht für das Datum der letzten Änderung der Datei, "changefreq" für die voraussichtliche Änderungsrate der Datei und "priority" für die Priorität dieser URL gegenüber den anderen, wobei hier Werte von 0.0 bis 1.0 gültig sind. (NP:Web27, vgl. sitemaps.org 07.01.2021)

Technischer Teil der Webseite

Pagespeed & Erreichbarkeit

Die Geschwindigkeit mit welcher die gesamte Webseite geladen ist und dem Benutzer angezeigt wird, spielt auch eine große Rolle bei der Bewertung. Eine Webseite, die schnell lädt, wirkt zum einen positiv auf den Benutzer und zum anderen auch positiv auf das Ranking der Seite. (NP:Web12, vgl. de.ryte.com 09.11.2020)

Weiters muss der Server kontinuierlich erreichbar sein, um überhaupt gecrawlt und indexiert zu werden. Webseiten die HTTPS¹¹ und damit SSL-Verschlüsselung für die Datenübertragung verwenden, bekommen automatisch eine bessere Wertung. (NP:Web12, vgl. de.ryte.com 09.11.2020)

¹¹Hypertext Transfer Protocol Secure

Mobile Optimierung

Seit dem “Google Mobile-Friendly Update”, welches 2015 von Google veröffentlicht wurde, zählt auch die Nutzerfreundlichkeit einer Webseite für Mobilgeräte zum Ranking. Demnach werden Seiten, welche für die Anzeige auf mobilen Geräten optimiert wurden, in den Suchergebnissen höher platziert. Dabei wirkt sich diese Optimierung nicht auf das Ranking für normale Suchanfragen aus, sondern nur für die Suchanfragen, welche über ein mobiles Gerät getätigt werden. Die mobile Optimierung muss nicht unbedingt für die ganze Webseite durchgeführt werden, sie kann auch nur für einzelne Unterseiten vorgenommen werden; Ersteres ist aber empfehlenswert. (NP:Web16, vgl. de.ryte.com 21.11.2020)

Fehler, die mit der mobilen Optimierung behoben werden müssen:

- Blockierte Inhalte
Die “robots.txt”-Datei sollte richtig konfiguriert sein; sprich, alle benötigten Java-Script-, CSS¹²- und Bilddateien müssen für den Crawler zugänglich sein.
- Nicht abrufbare Inhalte
Videos und Animationen müssen mit HTML5-Standards in die Webseite eingebunden werden, um deren richtige Darstellung zu gewährleisten.
- Weiterleitungs-Fehler
Auf mobil-optimierten Seiten, sollten nur Weiterleitungen auf Seiten bestehen, die selbst auch für mobile Geräte optimiert sind, ansonsten sollten Weiterleitungen verhindert werden.
- Spezifische 404-Fehler
Solche Fehler treten auf, wenn Content auf einer Webseite auf der Desktopversion angezeigt wird, auf der Mobilversion jedoch nicht. Diese Fehler sind dann nur auf mobilen Geräten sichtbar.
- ...

(NP:Web16, vgl. de.ryte.com 21.11.2020)

Weiterleitungen bei nicht vorhandenen Seiten

Wenn Unterseiten gelöscht werden oder innerhalb der Webseite verschoben werden, sodass sie nicht mehr unter derselben URL abrufbar sind, treten für den Benutzer auch 404-Fehler auf; insofern der Nutzer versucht, die alte URL aufzurufen. (NP:Web13, vgl. vioma.de 09.11.2020)

Aus diesem Grund müssen Redirects für die alten URLs eingesetzt werden, welche auf die neuen passenden URLs verweisen. So wird der User, wenn er eine jetzt ungültige

¹²Cascading Style Sheets

Adresse aufruft, auf die neue Seite umgeleitet und bekommt die gewünschten Inhalte trotzdem angezeigt. Die Suchmaschine überträgt das Ranking der alten Seite auf die neue Seite jedoch nur, wenn der Content der gleiche ist. Durch das Verwenden von Redirects kann man also verhindern, dass die Bewertung für eine Unterseite bei einer Umstrukturierung der gesamten Webseite verloren geht. (NP:Web13, vgl. vioma.de 09.11.2020)

2.6.2 Offpage-Optimierung

Bei der Offpage-Optimierung geht es darum, eine gewisse Bekanntheit für die Webseite aufzubauen, sprich sie beinhaltet Maßnahmen, die außerhalb der Webseite unternommen werden. Die Reputation einer Webseite wird hauptsächlich durch die Anzahl an den Backlinks (siehe Abschnitt 2.3) gemessen. Wenn viele verschiedene Seiten also auf die eigene Seite verweisen, wirkt sich dies beträchtlich positiv auf das Ranking der eigenen Seite aus. (NP:Web17, vgl. de.ryte.com 22.11.2020)

Seit Dezember 2010 wird vermutet, dass Social Signals, also die Erwähnung einer Webseite auf sozialen Netzwerken oder in Blogs (Social Media), geringfügig das Ranking beeinflussen. An erster Stelle steht allerdings noch immer das Linkbuilding, mit dem ein möglichst großes Netz aus Empfehlungen von anderen Webseiten über Backlinks erarbeitet wird. (NP:Web17, de.ryte.com 22.11.2020)

Linkbuilding

Die Offpage-Optimierung wird hauptsächlich durch gutes Linkbuilding vollzogen; Linkbuilding bedeutet, eine Anzahl an qualitativ hochwertigen und themenrelevanten Backlinks aufzubauen. Dadurch kann das Ranking der Webseite nachhaltig gesteigert werden und die Seite wird folglich in den Suchergebnissen besser platziert. Beim Linkbuilding muss natürlich auch einiges, welches in den folgenden zwei Abschnitten erläutert wird, beachtet werden. (NP:Web17, vgl. de.ryte.com 22.11.2020)

Quantitative Faktoren

Allgemein kann man sagen, je mehr Backlinks auf die Seite zeigen, desto höher ist auch die Popularität. Für den Aufbau dieser Linkbasis gibt es mehrere Methoden. Google empfiehlt als beste Methode "einfach nichts tun", denn wenn der Content auf der Webseite einen gewissen Mehrwert für den Benutzer bietet, dann werden diese die Webseite in Form von Backlinks auf ihrer eigenen Seite weiterempfehlen. (NP:Web17, vgl. de.ryte.com 22.11.2020)

Andere Methoden, wie etwa der reziproke Linktausch, wobei zwei Partner auf ihrer Webseite absichtlich auf einander verlinken, Blog-Kommentare, Linkkauf, Linkmiete und Einträgen in Webkatalogen sind mittlerweile nicht mehr empfehlenswert, da Google längst Mechanismen entwickelt hat, um dies zu erkennen. Werden Links also auf einem unnatürlichen Weg generiert, kann dies negativen Einfluss auf das Ranking haben, da die Webseite so gegen die Google-Richtlinien verstößt. (NP:Web17, vgl. de.ryte.com 22.11.2020)

Qualitative Faktoren

Die Backlinks, die im Rahmen des Linkbuildings aufgebaut werden, werden durch die Suchmaschine einzeln nach ihrer Qualität bewertet. Dabei gilt lieber "Qualität statt Quantität" und weniger ist hierbei häufig mehr. Der Grund dafür ist, dass wenige, aber dafür gute Links allgemein besser bewertet werden. Nachstehend sind einige Faktoren, die in die Bewertung eines einzelnen Backlinks einfließen. (NP:Web17, vgl. de.ryte.com 22.11.2020)

- Themenrelevanz

Wenn durch Algorithmen erkannt wird, dass der thematische Bezug von der verlinkenden zur verlinkten Seite fehlt, werden diese betroffenen Backlinks automatisch schlechter bewertet als die, bei denen ein Bezug besteht.

- Anchor Text

Dies ist der Text, der für den Benutzer als klickbarer Link sichtbar ist. Dieser Text ist auch ein ausschlaggebendes Kriterium. Damit die Suchmaschine einen Bezug zum Ziel des Links herstellen kann, sollte im Linktext ein Keyword (über welches die Zielseite gefunden werden will) vorkommen. Hierbei muss aber auch aufgepasst werden, denn solche Verlinkungen könnten eventuell als unnatürlich interpretiert werden und somit schlechter bewertet werden.

- Position auf der Webseite

Einfluss auf die Bewertung eines einzelnen Links hat auch die Position dieses auf der Webseite. Zum Beispiel werden Links, die im Header platziert werden, extrem schlecht bewertet oder gar nicht für das Ranking herangezogen. Hingegen Links, welche im Content der Webseite und in einem inhaltlichen Zusammenhang stehen, werden sehr gut bewertet.

- "no-follow"-Attribut

Wenn das "a"-Tag (siehe Abschnitt 2.3), mit dem der Link auf der Seite eingebunden wird, das rel="nofollow"-Attribut beinhaltet, dann ist dieser Link ziemlich wertlos, da die Suchmaschine diesen nicht zur Indexierung verwendet und somit auch das Ranking der Webseite dadurch nicht beeinflusst wird.

(NP:Web17, vgl. de.ryte.com 22.11.2020)

Kapitel 3

Crawler

3.1 Überblick

Im folgenden Kapitel wird näher auf die eigentlichen Bestandteile und die Arbeitsweise eines Webcrawlers eingegangen. Zudem werden die Arten und Einsatzgebiete für Crawler erklärt, sowie Probleme für Crawler beschrieben. Zuletzt werden Frameworks zur Programmierung eines eigenen Webcrawlers aufgezeigt.

3.2 Anforderungen an einen Webcrawler

Die Anforderungen, die an einen Crawler gestellt werden, können in Muss- und Soll-Anforderungen geteilt werden. (NP:Article03, vgl. Manning et al. 2009, S. 443-444)

Muss-Anforderungen:

- Robustness
Crawler treffen beim Crawlen des Internets auf sogenannte “Spider-Traps”, meist von Servern generierte unendliche Strukturen, bei denen der Crawler eine unendliche Anzahl von Links innerhalb einer Domain crawlt und dort sozusagen “gefangen” ist. Manche dieser Traps werden absichtlich erstellt, damit der Crawler nicht zu anderen Bereichen einer Seite kommt, oder andere Seiten im Web besucht. (NP:Web22, vgl. techopedia.com 02.01.2021)

Anders können solche Fallen auch unbeabsichtigt entstehen, beispielsweise wenn Kalender-Objekte auf einer Webseite implementiert werden und das ausgewählte Datum der URL zugefügt wird. So kann es dazu kommen, dass auch Daten, die in der Zukunft liegen, als gültig akzeptiert werden und so auch unendlich viele Links entstehen. (NP:Web22, vgl. techopedia.com 02.01.2021); Mehr Informationen zu "Spider-Traps" in Abschnitt 3.9.

- **Politeness**

Ein Crawler sollte sowohl explizite als auch implizite Richtlinien, wie er Webseiten zu besuchen hat, beachten. Implizite Richtlinien sind dabei die Regeln, die vom Webserver in der robots.txt-Datei (siehe 3.7) festgelegt werden. Explizite Regeln werden hingegen von niemandem festgelegt → diese sind eher selbstverständlich. Beispielsweise sollte beachtet werden, nie zu viele Anfragen aufgrund von Überlastung an denselben Webserver zu schicken.

(NP:Article03, vgl. Manning et al. 2009, S. 443-444); (NP:Article04, vgl. Herta 2009, S. 4-6)

Soll-Anforderungen:

- **Distributable**

Der Crawler sollte auf mehreren verschiedenen Maschinen gleichzeitig synchronisiert ausführbar sein.

- **Scalability**

Die gewählte Crawler-Architektur sollte es ermöglichen, die Crawl-Rate, also wie viele Seiten pro Minute gecrawlt werden können, durch Hinzufügen zusätzlicher Maschinen und Netz-Bandbreite zu skalieren.

- **Performance and efficiency**

Systemressourcen wie Prozessor, Speicher und Netzwerkbandbreite sollten effizient vom Crawler genutzt werden.

- **Quality**

Da nur ein geringer Teil aller Webseiten im Internet für den User nützlich sind, sollte der Crawler darauf ausgerichtet sein nützliche Seiten zuerst zu crawlen.

- **Freshness**

Crawler laufen in vielen Anwendungsfällen kontinuierlich; sie crawlen jede Seite nach einer bestimmten Zeitperiode nochmals. Bei Suchmaschinen wird so sicher gestellt, dass indizierte Webseiten immer wieder aktualisiert werden und mit ihrem neuesten Stand im Index abgespeichert sind. Damit dies funktioniert, sollte der Crawler in der Lage sein, eine Seite mit einer Frequenz zu crawlen, die in etwa ihrer Änderungsrate entspricht.

- Extensibility

Damit ein Crawler leicht erweiterbar ist und auf neue (Netzwerk-) Protokolle und Datenformate angepasst werden kann, sollte die Architektur modular aufgebaut werden.

(NP:Article03, vgl. Manning et al. 2009, S. 443-444); (NP:Article04, vgl. Herta 2009, S. 4-6)

3.3 Komponenten eines Crawlers

Crawler bestehen aus 3 Hauptkomponenten: dem Frontier, dem Page Downloader und dem Web Repository. Diese Architektur ist allerdings der simple allgemeine Aufbau; es gibt auch andere Modelle, bei denen das Frontier in mehrere kleine Module aufgeteilt wird, die zusammen die Aufgaben des Frontiers übernehmen. Das gleiche gilt auch für den Page Downloader, welcher etwa genauer in beispielsweise einen URL-Filter, Content-Filter oder Parser aufgeteilt werden kann. (NP:Article01, vgl. S.AMUDHA et al. 2017, S. 133); (NP:Article02, vgl. Drescher 2010, S. 13-15); (NP:Article03, vgl. Manning et al. 2009, S. 443-444)

- **Crawler Frontier**

Das Crawler Frontier beinhaltet eine Liste von Seed-URLs, welche vom Crawler im Laufe seines Arbeitsvorganges besucht werden. Der Inhalt der Liste wird dabei entweder statisch festgelegt, oder von einem User oder einem anderen Programm übergeben. Im Frontier wird auch festgelegt, welche Richtlinien und welche Logik der Crawler beim Besuchen der einzelnen Seiten befolgt. Jede URL bekommt eine bestimmte Priorität zugewiesen, um festzulegen, welche Links vom Crawler zuerst besucht werden sollen. Je nach Strategie, welche das Programm verfolgt, wird die Priorität einer URL anders ermittelt. (NP:Article01, vgl. S.AMUDHA et al. 2017, S. 133)

Zum einen kann sie vom Auftreten von bestimmten Keywords auf der Seite berechnet werden, oder sie ist beispielsweise abhängig von der Änderungsrate oder der Qualität der Webseite. Seiten, die aktiv aktualisiert und erneuert werden, sowie Seiten, die in den Suchergebnissen hoch geranked werden (ein gutes Ranking impliziert Qualität), werden bevorzugt. (NP:Article01, vgl. S.AMUDHA et al. 2017, S. 133); Für mehr Informationen bezüglich des Rankings und Qualität einer Webseite siehe 2.6.

- **Page Downloader**

Der Page Downloader ist für das Herunterladen der Webseite aus dem Internet und für das Parsen der Seite zuständig. Dazu ist er als simpler HTTP-Client aufgebaut, der an die vom Frontier bereitgestellten URLs HTTP-Requests stellt. Außerdem muss er die HTML-Seite vom HTTP-Response auslesen und Hyperlinks auf der Seite herausfiltern, normalisieren und diese Links dann wieder an das Frontier schicken. (NP:Article02, vgl. Drescher 2010, S. 15)

Die Normalisierung der URLs erfolgt nach dem RFC 3986¹ Standard:

- Einheitliche Kleinschreibung

Zur Normalisierung von URLs wird allgemein die ganze URL in lowercase, sprich Kleinbuchstaben geschrieben.

http://www.beiSPIEL.at/
wird zu
http://www.beispiel.at/

- Prozent-codierte Zeichen in Großbuchstaben umwandeln

Hexadezimale Ziffern, die entstehen, wenn besondere Zeichen innerhalb der URL codiert werden, werden normalisiert, indem sie großgeschrieben werden.

http://www.beispiel.at/foo%2a
wird zu
http://www.beispiel.at/foo%2A

- Dekodieren von prozent-codierten nicht reservierten Zeichen

Prozent-codierte Zeichen in den Bereichen ALPHA (%41 - %5A und %61 - %7A) und DIGIT (%30 - %39) sowie Bindestriche (%2D), Punkte (%2E), Unterstriche (%5F) und Tilde (%7E) erfordern die Prozent-Codierung nicht und sollten deswegen dekodiert werden. Für eine Liste aller prozent-codierten Zeichen siehe (NP:Web20, key-shortcut.com 30.12.2020).

http://www.beispiel.at/%7Efoo
wird zu
http://www.beispiel.at/~foo

¹<https://tools.ietf.org/html/rfc3986>

- Punktsegmente entfernen

Innerhalb der URL haben “./” und “../” spezielle Bedeutungen, sie referenzieren auf das aktuelle Verzeichnis bzw. das Parent-Verzeichnis. Die URL wird durch das Weglassen dieser speziellen Segmente vereinfacht.

```
http://www.beispiel.at/a/./b/.../c  
wird zu  
http://www.beispiel.at/a/c
```

- Abschließen mit einem “/”

Wenn nach der Autoritätskomponente oder nach einem Verzeichnis kein “/” folgt, wird eines angehängt:

```
http://www.beispiel.at  
wird zu  
http://www.beispiel.at/
```

bzw.:

```
http://www.beispiel.at/foo  
wird zu  
http://www.beispiel.at/foo/
```

- Entfernen des Standardports

Der Port (falls Standardport) inklusive dem Trennzeichen “:” wird aus der URL gelöscht:

```
http://www.beispiel.at:80/  
wird zu  
http://www.beispiel.at/
```

- Entfernen von Standarddateien

Standarddateien sind nicht benötigt und werden deswegen aus der URL gelöscht.

```
http://www.beispiel.at/index.html  
wird zu  
http://www.beispiel.at/
```

- Entfernen des Ankers

Ankerpunkte innerhalb einer Seite werden aus der URL gelöscht.

`http://www.beispiel.at/page.html#main`

wird zu

`http://www.beispiel.at/page.html`

- Entfernen doppelter Schrägstriche

Wenn innerhalb einer URL zwei benachbarte Schrägstriche vorkommen, werden diese mit nur einem Schrägstrich ersetzt.

`http://www.beispiel.at/foo//page.html`

wird zu

`http://www.beispiel.at/foo/page.html`

- Entfernen oder Hinzufügen von “www”

Manche Webseiten sind über zwei Internetdomänen verfügbar: über eine, bei der das “www” weggelassen wird und über eine, die als niedrigstwertiges Label “www” hat. Beide URLs liefern dann dieselbe Webseite.

`http://www.beispiel.at/`

wird zu

`http://beispiel.at/`

bzw.:

`http://beispiel.at/`

wird zu

`http://www.beispiel.at/`

- Sortierung von Query-Variablen

Bei mehr als einer Query-Variable innerhalb der URL werden die Parameter in alphabetischer Reihenfolge sortiert:

`http://www.beispiel.at/page?lang=de&article=crawler`

wird zu

`http://www.beispiel.at/page?article=crawler&lang=de`

(NP:Web18, NP:Web19, vgl. audisto.com, de.qaz.wiki 29.12.2020)

Nach der Normalisierung können auch Links, die zuvor schon besucht wurden, wieder auftreten. Je nachdem, welche Einstellungen im Frontier getroffen werden, werden gleiche URLs nicht mehr besucht oder nach einer bestimmten Zeit nochmals gecrawlt. (NP:Article01, vgl. S.AMUDHA et al. 2017, S. 133);

Für den HTTP-Client kann eine Timeout-Periode eingestellt werden, damit festgelegt werden kann, nach welcher verstrichenen Zeit die Abfrage für eine bestimmte URL abgebrochen werden soll. So wird sichergestellt, dass nicht unnötig lange Zeit verbraucht wird, um große Dateien zu lesen oder auf eine Antwort von einem langsamen Server zu warten. (NP:Article01, vgl. S.AMUDHA et al. 2017, S. 133);

- **Web Repository**

Die vom Page Downloader runtergeladenen Webseiten werden im Web Repository, einer Datenbank abgespeichert. Die gesamte HTML-Seite (Text und Metadaten, wie zum Beispiel Title-Tag und Meta-Description) wird hier gespeichert. Die Aufgaben des Web Repository umfassen beispielsweise das Speichern und Aktualisieren dieser Seiten. (NP:Article01, vgl. S.AMUDHA et al. 2017, S. 133)

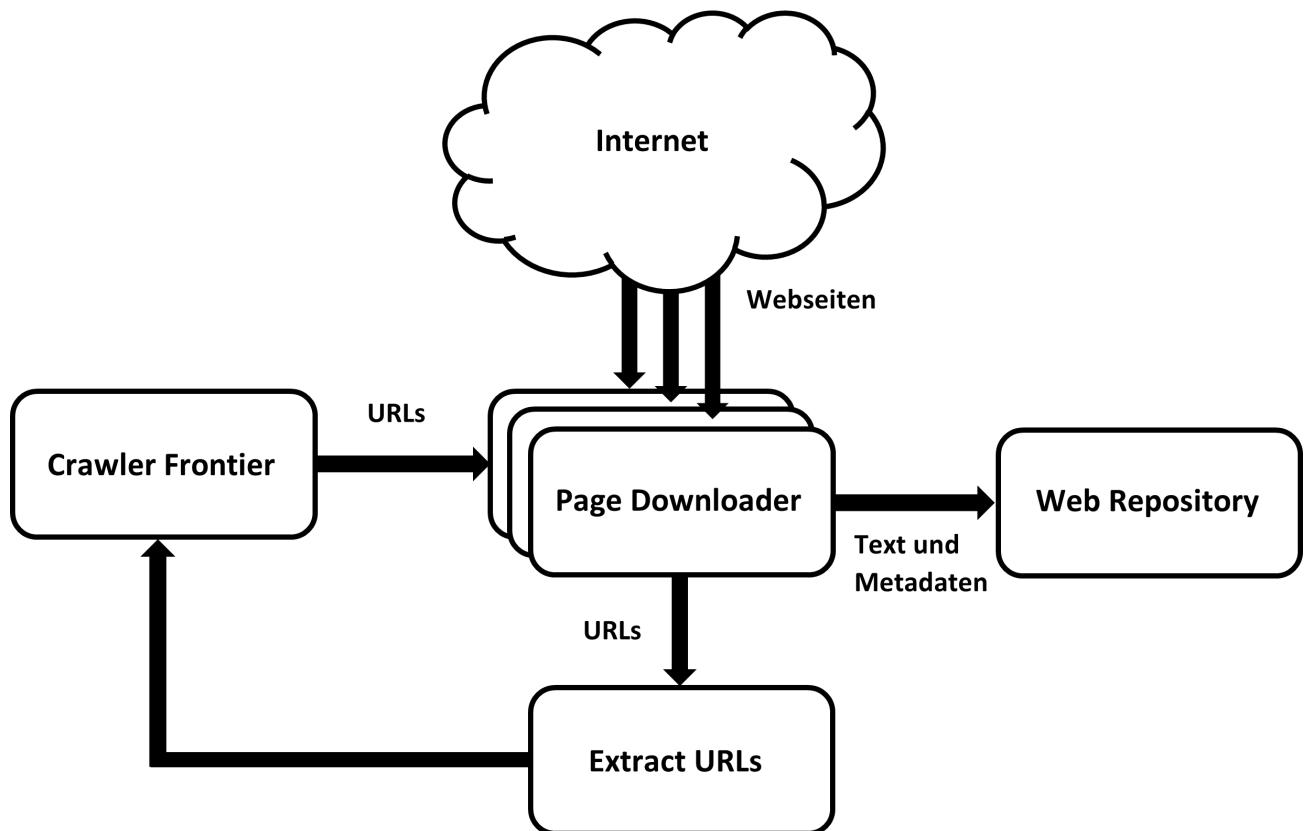


Abbildung 3.1: Architektur eines Crawlers
Angelehnt an: (NP:Article01, S.AMUDHA et al. 2017, S. 133)

3.4 Arbeitsprozess eines Crawlers

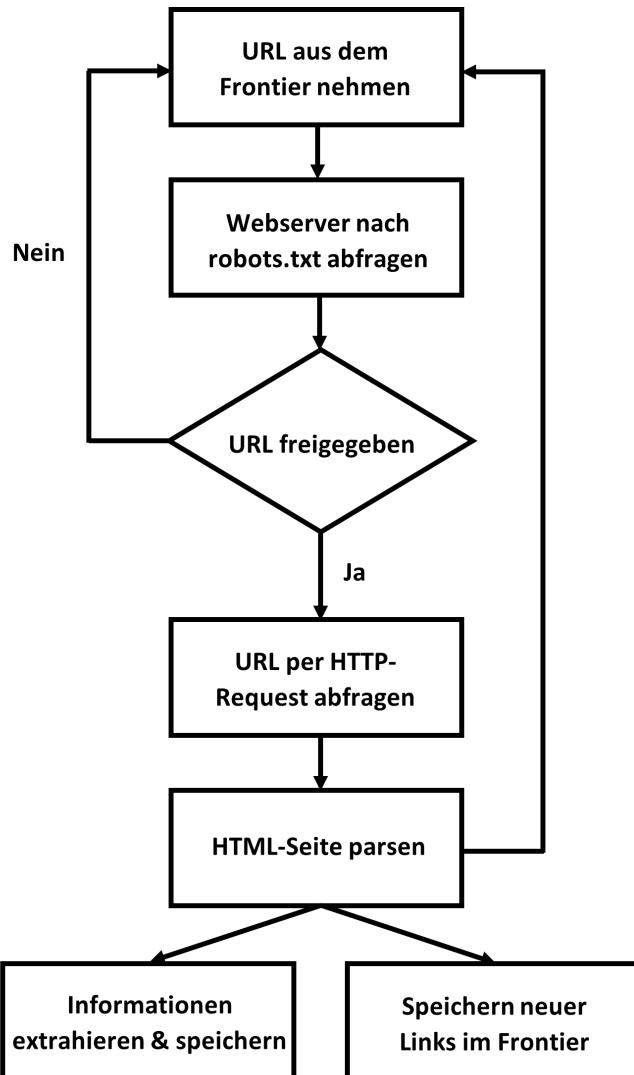


Abbildung 3.2: Arbeitsprozess eines Crawlers
 Quelle: Angelehnt an (NP:Article01, vgl. S.AMUDHA et al. 2017, S. 130)

Der Arbeitsprozess des Crawlers startet damit, dass eine Seed-URL aus dem Frontier genommen wird. Das Auswählen einer URL erfolgt nach einem Prioritätsprinzip → je höher die Priorität eines Links, desto eher wird dieser zuerst besucht.

Der Page Downloader stellt an die übergebene URL einen HTTP-Request, liest die robots.txt (siehe 3.7) für die jeweilige Domain aus und speichert diese Datei im Cache. Bei einem entsprechenden Response (200 OK, 201 Created) bekommt er die HTML-Seite der URL zurückgeliefert. Wenn die angefragte URL durch die robots.txt-Datei

freigegeben ist, wird die HTML-Seite geparsed, sprich, sie wird nach Hyperlinks (siehe 2.3) untersucht und bestimmte Informationen werden extrahiert.

Das Herausfiltern neuer Links wird entweder mit einer Regular Expression, einem entsprechendem HTML-Parser (Beautiful Soup [Python], Jericho HTML Parser [Java]) oder mit XPath (XML Path Language) realisiert. Gefundene Links müssen anschließend vom Page Downloader normalisiert werden → zwei unterschiedliche Links, welche sich in ihrer Syntax unterscheiden, aber auf dieselbe Seite verweisen, sollten nicht beide gecrawlt werden und deshalb auch nicht doppelt im Frontier gespeichert werden.

Nachdem ein Link besucht wurde, wird er aus der Liste im Frontier gelöscht und die gesamte HTML-Seite und herausgefilterte Informationen werden im Web Repository abgespeichert. Im Zuge des Crawlingvorganges können schon besuchte Links wieder gefunden werden. Je nachdem welche Einstellungen im Frontier getroffen werden, werden diese URLs wieder in die Liste der Seed-URLs geschrieben und somit nochmals gecrawlt.

Dieser Vorgang wird solange wiederholt, bis keine neuen URLs mehr vorhanden sind, oder eine andere Bedingung den Crawler stoppen lässt. Solch eine Bedingung könnte eventuell ein HTTP-Fehlercode beim Abrufen einer URL sein, oder ein Timeout, also falls nach einer bestimmten Zeitperiode keine Antwort vom Server kommt, sein.

(NP:Article01, vgl. S.AMUDHA et al. 2017, S. 133); (NP:Article02, vgl. Drescher 2010, S. 15-16); (NP:Article03, vgl. Manning et al. 2009, S. 445-448)

3.5 Crawlingstrategien

3.5.1 Breadth-First & Depth-First

Bei dem Breadth-First-Search beinhaltet das Frontier nur eine kleine Menge an Seed-URLs, mit denen der Crawler zu arbeiten beginnt. Weiters folgt er den gefundenen Links nach der “Breadth-First”-Methode, also First-in-First-out (FIFO) → er folgt zuerst denen, die auf derselben Ebene liegen. (NP:Web33, vgl. open4tech.com 03.02.2021)

Das Gegenteil dazu ist der Depth-First-Search: Das Frontier beinhaltet wieder nur ein kleines Set an URLs, doch diesmal folgt der Crawler den Links nach der “Depth-First”-Methode, Last-in-First-out (LIFO) → also nimmt er die Links mit der tiefsten Ebene zuerst, bevor er die anderen Links besucht. (NP:Web33, vgl. open4tech.com 03.02.2021)

In der nachstehenden Grafik sind beide Strategien veranschaulicht.

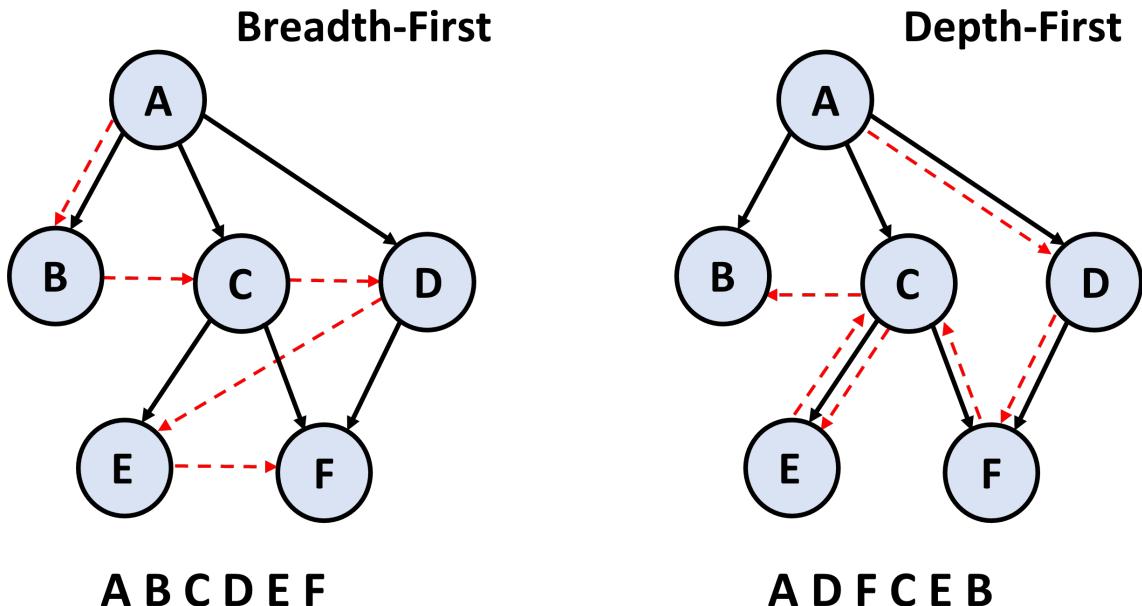


Abbildung 3.3: Breath-First-Search vs Depth-First-Search
Angelehnt an: (NP:Web33, open4tech.com 03.02.2021)

3.5.2 Incremental Crawler

Ein incremental Crawler durchläuft kontinuierlich das gesamte Web und aktualisiert das Set an zu crawlenden URLs laufend, anstatt jedesmal einen neuen Prozess zu starten. Dabei wird ein Modell verwendet, mit dem festgestellt werden kann, ob sich eine Seite seit dem letzten Besuch verändert hat und ob die Aktualisierungen neu erfasst werden müssen. Dafür werden Daten aus früheren Durchläufen verwendet um aktuelle Ergebnisse zu liefern. Statische Seiten werden dadurch nicht oft gecrawlt und die Ressourcen für wichtige Seiten verwendet. (NP:Article05, vgl. Mini et al. 2014, S. 133)

3.5.3 Focused Crawler

Ein Focused Crawler oder auch Topic Crawler ist darauf ausgelegt, nur Seiten welche zu einem bestimmten Thema passen zu besuchen. So sollen Zeit, Speicher sowie Netzwerkressourcen gespart werden. Solch ein Crawler besitzt zusätzliche Module; den Classifier und den Distiller. Ersterer ist dafür zuständig, dass der Content, der auf

einer Seite gefunden wird, mit dem vorher festgelegten Thema verglichen wird. Wenn die Inhalte evaluiert werden und einen gewissen Grenzwert überschreiten → somit als relevant angesehen werden, dann wird diese Seite weiterverarbeitet, ansonsten ignoriert. (NP:Article06, vgl. Linxuan et al. 2020, S. 3)

Weiters sorgt der Distiller dann für die Einschätzung der Relevanz dieser Seiten und, dass nur Seiten mit einer hohen Priorität weiterverarbeitet werden. (NP:Article08, vgl. Risha et al. 2014, S. 1)

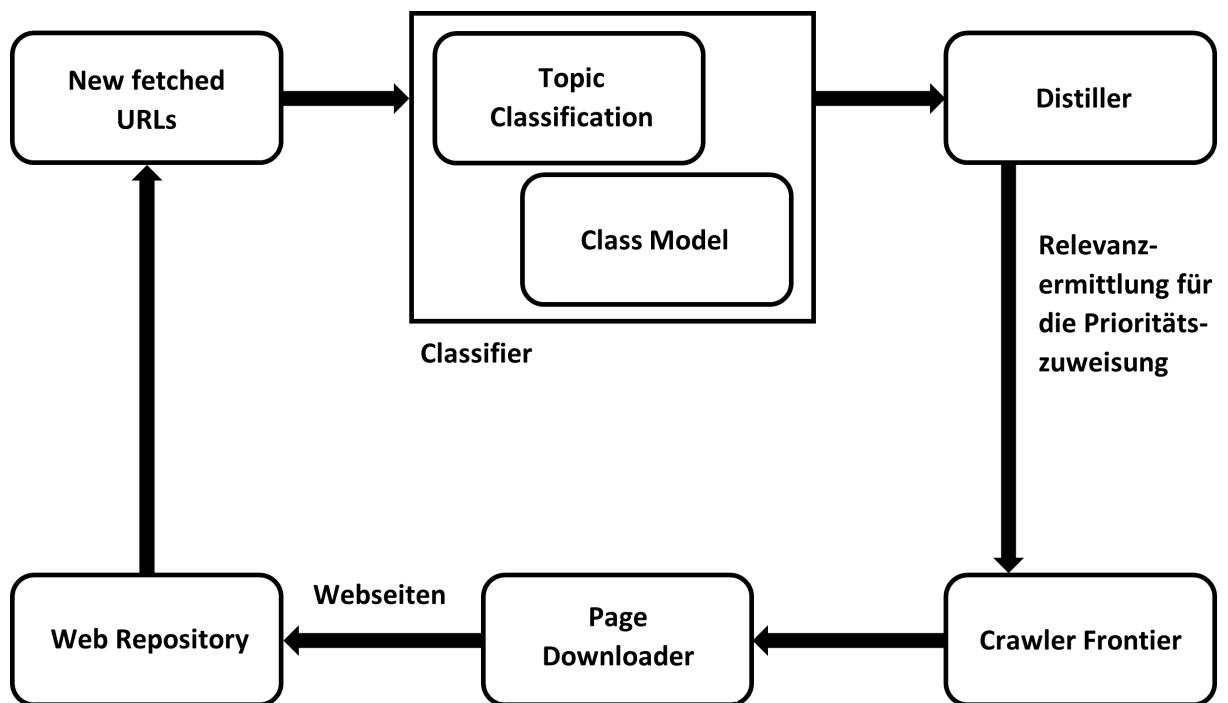


Abbildung 3.4: Architektur eines Focused Crawlers
Angelehnt an: (NP:Article08, vgl. Risha et al. 2014, S. 1)

3.5.4 Distributed Crawler

Distributed Crawlers werden in einem Netzwerk von vielen Maschinen betrieben. Ziel ist es, viele Seiten in einer angemessenen Zeitspanne zu crawlern. Durch solch ein Netzwerk ist es einfach den Gesamtprozess zu skalieren. Zusätzlich werden die Prozesse, die auf einer Maschine laufen, parallelisiert, um den Durchsatz zu erhöhen. (NP:Article05, vgl. Mini et al. 2014, S. 134)

Ein Gerät im Netz fungiert als Master und verteilt alle zu crawlenden URLs an die im Netzwerk betriebenen Crawler. Heruntergeladene Seiten werden an einen zentralen Indexer geschickt, welcher die Links extrahiert, die dann wiederum vom Master verteilt werden. So kann Downloadgeschwindigkeit und Zuverlässigkeit erhöht werden. (NP:Article05, vgl. Mini et al. 2014, S. 134)

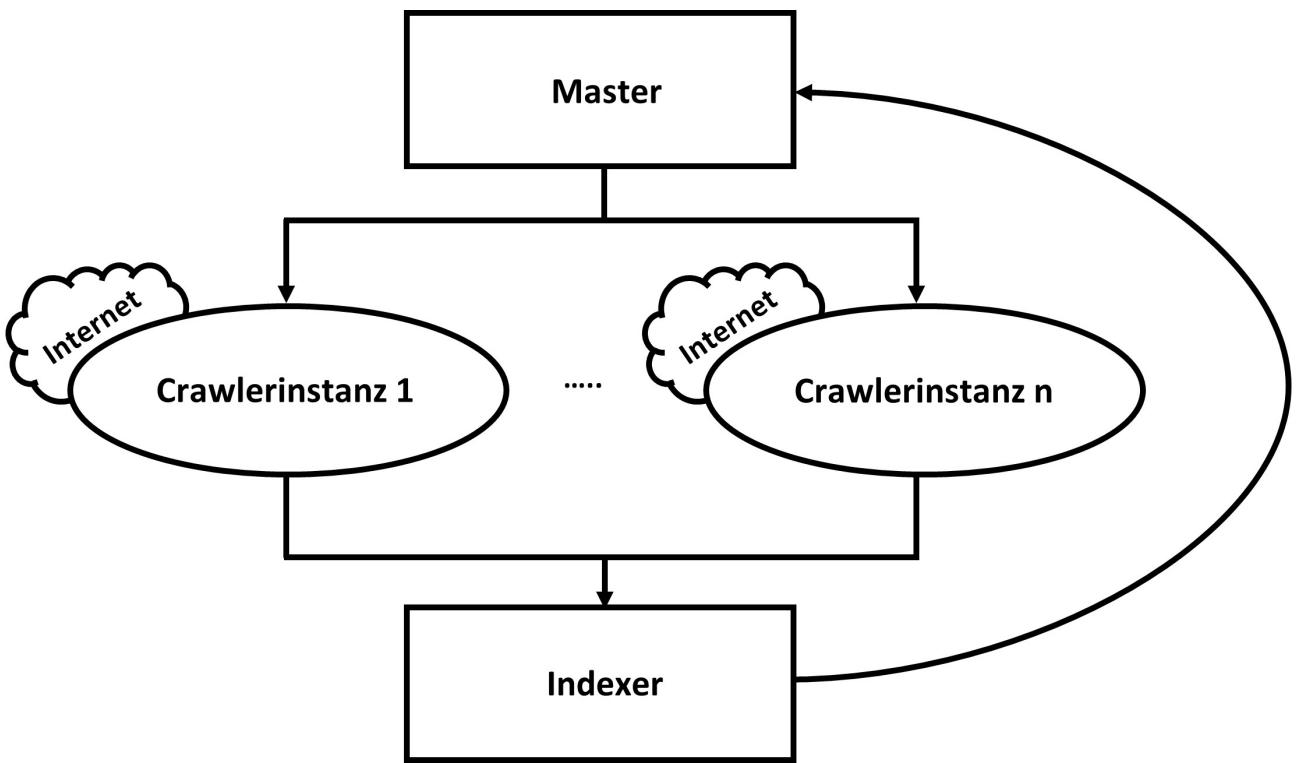


Abbildung 3.5: Architektur eines Distributed Crawlers

3.6 Anwendungsbereiche für Crawler

Wie auch schon am Anfang der Arbeit erwähnt, verwenden Suchmaschinen Crawler um einen Index aufzubauen, um so dem Benutzer bei Eingabe von bestimmten Keywords nützliche Suchergebnisse zu liefern. Mithilfe dieser Programme werden die Datenbanken am neuesten Stand gehalten und somit erscheinen keine veralteten Seiten in der Suche.

Weiters können Crawler auch zum Automatisieren von Wartungsarbeiten auf Webseiten eingesetzt werden. Dabei übernimmt er die Arbeit, Links innerhalb der Seite oder den HTML-Code zu validieren. (NP:Article05, vgl. Mini et al. 2014, S. 134)

Linguisten verwenden Webcrawler um textuelle Analysen durchzuführen → Durch den Einsatz ermitteln sie, welche Wörter heute häufig verwendet werden. (NP:Article05, vgl. Mini et al. 2014, S. 134)

Crawler wurden auch schon in biomedizinischen Anwendungen eingesetzt, um relevante Literatur zu Genen im Internet zu finden. (NP:Article07, vgl. Srinivasan et al. 2002, S. 1-8)

Webcrawler können auch für das sogenannte “Data-Mining” verwendet werden → Daten werden für die Marktanalyse, um Trends in bestimmten Bereichen zu erkennen, erfasst. Außerdem werden öffentliche E-Mail-Adressen (E-Mail-Harvesting) oder Telefonnummern zwecks Spam mit Crawlern ausgelesen. (NP:Article05, vgl. Mini et al. 2014, S. 134)

Preisvergleichsportale, wie beispielsweise <https://geizhals.at/>, <https://www.idealoo.at/> oder <https://www.preisvergleich.at/> nutzen dieselbe Technik, um Produktdaten und Preise von verschiedenen Anbietern für ein bestimmtes Produkt darzustellen.

The screenshot shows the product page for the Apple iPhone 12 128GB Black on idealo.at. At the top, there is a navigation bar with categories like Elektroartikel, Telekommunikation, Handys & Smartphones, and Apple iPhone 12. The main product image is displayed with a price range of € 794,00 – € 1,088,43. A 'weitere Varianten, z.B.: 64GB White' button is shown. To the right, a 'Preisentwicklung' (price development) chart shows the price trend from December 2020, with a current price of € 869,00. There are buttons for 3M, 6M, and 1J. Below the main product info, there are filters for Farbe (Color), interner Speicher (internal storage), and Zustand (condition). The 'Farbe' filter has 'schwarz' selected. The 'interner Speicher' filter has '128 GB' selected. The 'Zustand' filter has 'Neu ab € 794,00' selected. In the 'Preisvergleich' section at the bottom, two offers are listed: one from eBay for € 794,00 and one from Mobileshop for € 872,00. Both offers include payment methods like PayPal, VISA, and Mastercard, and delivery options like GLS.

Abbildung 3.6: User-Interface von idealo.at
Quelle: <https://www.idealoo.at/>

Bei der Suche nach Plagiaten werden Crawler ebenfalls verwendet, da riesige Datenmengen nach den gleichen Inhalten abgesucht werden können. Solch einen Dienst vertreibt zum Beispiel <https://www.turnitin.com/>. (NP:Web34, vgl. vabelhavt.at 04.02.2021)

Crawler kommen auch für die Web-Archivierung, also das Sammeln und dauerhafte Ablegen von Webseiten zum Einsatz. Sie erstellen Momentanaufnahmen der Seiten, die in Zukunft einen Blick in die Vergangenheit bieten sollen. (NP:Web34, vgl. vabelhavt.at 04.02.2021)

3.7 Robots Exclusion Standard

3.7.1 Allgemein

Das sogenannte "Robots Exclusion Standard Protokoll" (kurz: REP) wurde 1994 veröffentlicht. Dieses Protokoll legt fest, dass Suchmaschinen-Robots zunächst im Root-Verzeichnis nach einer Datei mit der Bezeichnung robots.txt suchen und die enthaltenen Vorgaben auslesen, bevor sie mit der Indexierung beginnen. (NP:Web24, de.ryte.com 04.01.2021)

Mithilfe der robots.txt-Datei, die auch schon mehrere Male im Zuge dieser Arbeit erwähnt wurde, können Webseitenbetreiber das Verhalten von Crawlern, die die Seiten für eine Suchmaschine indexieren, innerhalb ihrer Domain steuern. Zwar kann man mit Meta-Tags auf einer HTML-Seite auch festlegen ob diese zur Indexierung verwendet werden soll, jedoch gilt dies nur für diese eine Datei und maximal alle darin durch Verweise erreichbaren Seiten. (NP:Web23, vgl. wiki.selfhtml.org 04.01.2021)

Mit der robots.txt werden Regeln für ganze Verzeichnisse unabhängig der Verzeichnistruktur aufgesetzt. Für den Robots Exclusion Standard gibt es kein geschriebenes RFC², weshalb die Syntax von den Crawlern nicht immer einheitlich gehandhabt wird. Deswegen ist es empfehlenswert, Meta-Tags zu verwenden, um die gewünschte Indexierung der Seiten zu gewährleisten. (NP:Web23, vgl. wiki.selfhtml.org 04.01.2021)

Mit der Datei kann man den Zugriff von Crawlern oder Personen auf die Seiten, welche nicht gecrawlt und damit auch nicht in den Suchergebnissen erscheinen sollen, nicht einschränken. In der Regel halten sich Suchmaschinencrawler an diese Regeln, Crawler mit bösen Absichten halten sich jedoch nicht an die Vorgaben der robots.txt. (NP:Web23, vgl. wiki.selfhtml.org 04.01.2021)

²Request For Comments

3.7.2 Speicherort

Die Datei ist eine einfache Textdatei und kann mit jedem Texteditor erstellt und bearbeitet werden; sie wird im Plaintext-Format abgespeichert und ausgelesen. Wie vorhin schon angemerkt, wird die robots.txt im Wurzelverzeichnis der Web-Dateien für eine Domain gespeichert. Alle Zeichen im Dateinamen müssen dabei kleingeschrieben werden und es kann pro Domain nur eine robots.txt-Datei geben. (NP:Web24, vgl. de.ryte.com 04.01.2021)

Beispielsweise ist der Pfad zur Datei für die Domain “beispiel.at” → “<https://beispiel.at/robots.txt>”. Es ist wichtig, dass die Datei keine syntaktischen Fehler aufweist und generell fehlerfrei ist. Auch kleine Fehler können dazu führen, dass die eigentlich vorgesehenen Beschränkungen nicht beachtet werden. Ein hilfreiches Tool, den “robots.txt-Tester”, stellt dafür die Google Search Console³ zur Verfügung, mit dem man die Datei überprüfen kann. (NP:Web24, vgl. de.ryte.com 04.01.2021)

3.7.3 Aufbau

Die robots.txt wird aus Datensätzen, sogenannten “records” aufgebaut. Ein solcher Eintrag besteht allgemein aus zwei Teilen. Im ersten Teil, dem User-agent, wird festgelegt, für welche Crawler die danach im zweiten Teil folgenden Regeln gelten.

Regeln werden mit den Keywords “Allow” und “Disallow” beschrieben, wobei Ersteres nur in seltenen Fällen verwendet wird. Einzelne Records werden mit einer Leerzeile getrennt. Leerzeilen innerhalb eines Records sind ungültig. (NP:Web23, vgl. [wiki.selfhtml.org 04.01.2021](https://en.wikipedia.org/wiki/Robots_exclusion_standard))

Eine simple robots.txt sieht beispielsweise so aus:

```
1 User-agent: Googlebot
2 Disallow:
```

→ der Googlebot darf laut obigem Beispiel somit alle Seiten crawlen; um ihm dies zu verbieten müsste die Datei folgendermaßen aussehen:

```
1 User-agent: Googlebot
2 Disallow: /
```

³Analysetool zur Suchmaschinenoptimierung

Mögliche Eingaben in der robots.txt (NP:Web25, mindshape.de 05.01.2021):

Eingabe	Erklärung	Beispiel
#	Kommentarzeile	# robots.txt für Beispiel.at
*	Wildcard; für User-agents und URLs	Disallow: /*?
\$	Kennzeichnet Pfadende	Disallow: /*.pdf\$
User-agent:	Angesprochene(r) Crawler	User-agent: Googlebot
Allow:	Erlaube Besuch (Default)	Allow: /beispiel.html
Disallow:	Verbiete Besuch	Disallow: /beispiel.html
Sitemap:	Angabe der Sitemap(s)	Sitemap: http://www.beispiel.at/index.php?id=xxx
Crawl-delay:	Verzögerung zwischen zwei Abrufen ⁴	Crawl-Delay: XXX
Noindex:	Entfernung angegebener Dateien aus dem Index (offiziell nicht unterstützt)	noindex: /beispiel.html

Tabelle 3.1: Mögliche Eingaben in der robots.txt

User-agent

Ein Record beginnt immer mit dem User-agent, dem Namen, unter dem sich der Crawler beim Webserver „anmeldet“ und bekannt gibt, er möchte die Webseite besuchen. Als User-agent kann immer nur ein Name angegeben werden; falls man aber eine Regel für mehrere Robots geltend machen will, können mehrere Zeilen, die mit User-agent beginnen, untereinander stehen. (NP:Web23, vgl. wiki.selfhtml.org 04.01.2021)

Der User-agent ist nicht case-sensitive, es ist also egal ob groß- oder kleingeschrieben wird. Mit einem Record können auch mehrere Crawler angesprochen werden, in dem man für den Namen den Platzhalter „*“ verwendet → die Regel trifft dann auf jeden User-agent zu. Mehr als ein Datensatz pro User-agent ist nicht erlaubt. (NP:Web23, vgl. wiki.selfhtml.org 04.01.2021)

```

1 User-agent: *
# Regeln 1
2
3
4 User-agent: Google
# Regeln 2
5
6 User-agent: Beispiel
# Regeln 3
7
8

```

⁴in Minuten

Der Name beim User-agent stellt nur den Anfang des User-agent-Strings dar, deswegen hätte User-agent: Google die gleiche Bedeutung wie User-agent: Google*. Der Crawler muss den Record bestimmen, für den sein User-agent die genaueste Übereinstimmung liefert. Die Reihenfolge der Datensätze innerhalb der robots.txt ist deshalb egal. (NP:Web23, vgl. wiki.selfhtml.org 04.01.2021)

Laut obigem Beispiel würde ein Robot mit dem User-Agent "TestRobot" die Regeln 1 verwenden, ein Robot mit dem Namen "Googlebot" die Regeln 2, und einer mit dem Namen "Beispielcrawler" die Regeln 3. (NP:Web23, vgl. wiki.selfhtml.org 04.01.2021)

Disallow

Nachdem eine Zeile mit User-agent begonnen wurde, folgt meist eine Zeile, die mit Disallow beginnt. Nach dem Disallow kann man einen Dateipfad innerhalb der Domain angeben, welcher vom vorhin genannten Crawler nicht besucht werden soll. Es ist möglich mehrere Disallow-Zeilen einzutragen, wobei die erste zutreffende Zeile genommen wird. (NP:Web23, vgl. wiki.selfhtml.org 04.01.2021)

```
1 User-agent: *
2 Disallow: /beispiel/
3 Disallow: /beispiel/bilder
```

In diesem Beispiel ist der zweite Disallow-Eintrag unnütz, da der Dateipfad schon in der vorhergehenden Zeile /beispiel/ vom Crawling ausgeschlossen wurde. Es ist wichtig, dass man den Dateipfad oder -namen richtig angibt, beispielsweise würde /bild genauso auch für /bilder/, /bilder/beispiel oder bild123.jpg gelten. Verzeichnisse sollte man also mit einem abschließenden "/" angeben. (NP:Web23, vgl. wiki.selfhtml.org 04.01.2021)

Allow

Ursprünglich gab es das Schlüsselwort Allow nicht, dies wurde erst 1996 eingeführt. Mit Allow kann man innerhalb von eigentlich gespererten Pfaden, die mit Disallow angegeben wurden, einzelne Verzeichnisse oder Dateien freigeben. Wenn keine Disallow-Regel den Zugriff auf ein Verzeichnis oder eine Datei verbietet, ist es nicht nötig, mit Allow diese ausdrücklich freizugeben. (NP:Web23, vgl. wiki.selfhtml.org 04.01.2021)

```
1 User-agent: *
2 Allow: /bilder/public/
3 Disallow: /bilder/
```

In obigem Beispiel wird das Verzeichnis /bilder/public/ freigegeben, jedoch alles andere was sich in /bilder/ befindet, wird von der Indexierung ausgeschlossen.

```
1 User-agent: *
2 Allow: /public/
3 Disallow: /
```

In diesem Beispiel wird alles innerhalb der Domäne für Crawler gesperrt, außer das Verzeichnis /public/ soll für die Indexierung verwendet werden.

Weiters muss beim Erstellen der robots.txt darauf geachtet werden, dass die Einträge der Reihe nach abgearbeitet werden, sprich Allow-Regeln haben keine Wirkung, wenn der Dateipfad vorhin durch ein Disallow verboten wurde. (NP:Web23, vgl. wiki.selfhtml.org 04.01.2021)

```
1 User-agent: *
2 Disallow: /bilder/
3 Allow: /bilder/public/
```

Demnach wird in obigem Beispiel das Verzeichnis /bilder/public/ trotz dem Allow-Eintrag nicht zur Indexierung verwendet.

Wildcards

Für die Verzeichnis-/Dateipfadangabe können die Wildcards “*” und “\$” verwendet werden; “*” steht dabei für beliebig viele Zeichen und “\$” für das Zeilenende (NP:Web23, vgl wiki.selfhtml.org 04.01.2021):

```
1 User-agent: *
2 Disallow: /bild*/    # alle Verzeichnisse, die mit "bild" beginnen
3 Disallow: /*bild*/  # alle Verzeichnisse, die das Wort "bild" enthalten
4 Disallow: /*.pdf$    # alle Dateien, die die Dateiendung ".pdf" haben
5 Disallow: /*?        # alle URLs, die Parameter enthalten
```

Häufige Fehler

- Fehlender Schrägstrich bei einer Verzeichnisangabe

Wenn ein bestimmtes Verzeichnis von der Indexierung ausgeschlossen werden soll, dann ist es wichtig, dass die Verzeichnisangabe mit einem "/" abgeschlossen wird. Mit "Disallow: /bilder" wird nicht nur /bilder/ gesperrt, sondern auch beispielsweise /bilder-public/.

- Unwissenliches Ausschließen beim Verwenden von Wildcards

Wenn man Wildcards verwendet, muss man unbedingt überprüfen, ob auch nur vorgesehene URLs durch diese betroffen werden → siehe Absatz zu Wildcards oben.

- Robots.txt statt robots.txt

Da der Crawler nach der "robots.txt" case-sensitive sucht, ist eine Datei mit dem Namen "Robots.txt" eine andere, als eine mit dem Namen "robots.txt".

- Zwei Verzeichnispfade in einer Zeile

Wenn zwei Verzeichnispfade gesperrt werden sollen, dann müssen diese in zwei aufeinanderfolgenden Disallow-Regeln eingetragen werden und nicht innerhalb einer Zeile stehen, da der Crawler die Eingabe nach einem Disallow als einzelnen Pfad liest.

→ Beispielsweise muss "Disallow: /public/ /admin/" in zwei Disallows aufgeteilt werden, damit dies die gewünschte Wirkung hat.

- Anweisungen betreffend Subdomains

Für jede Subdomain muss eine eigene robots.txt erstellt und abgelegt werden
→ die Regeln der Hauptdomain gelten hier nicht.

- robots.txt wird nicht im Root-Verzeichnis gespeichert

Wenn die robots.txt-Datei nicht im Root-Verzeichnis der Domain abgespeichert ist, dann wird sie vom Crawler nicht gefunden und der Crawler geht somit davon aus, dass jede Seite gecrawlt werden soll.

- Keine Leerzeile zwischen zwei Records

Damit Records gültig sind, muss nach dem Definieren von Regeln eine Leerzeile folgen, bevor die nächste User-agent Zeile folgt.

3.8 Beispiele für Suchmaschinencrawler

3.8.1 Googlebot

Googlebot ist der allgemeine Name für den Web-Crawler von Google. Genauer gesagt handelt es sich dabei um zwei verschiedene Arten von Crawlern: einen Computer-Crawler, der einen Nutzer auf einem Computer simuliert, und einen mobilen Crawler, der einen Nutzer auf einem Mobilgerät simuliert. (NP:Web21, developers.google.com 02.01.2021)

Jede Webseite wird von beiden Arten des Googlebots besucht. Im User-agent, der bei der Anfrage mitgeschickt wird, kann man den jeweiligen Bot identifizieren. Mit Einstellungen in der robots.txt-Datei ist es aber nicht möglich einen von den beiden gezielt anzusprechen. (NP:Web21, vgl. developers.google.com 02.01.2021)

Der Crawler wird von Google verwendet, um neue und aktualisierte Webseiten dem Google-Index hinzuzufügen. Der Index ist eine riesige Datenbank, in der alle nötigen Informationen zu den Webseiten abgespeichert werden. Sobald eine Seite im Index gespeichert ist, können Benutzer diese über die Suchmaschine finden. (NP:Article09, vgl. Imhof 2019, S. 19-20)

Der Googlebot wird gleichzeitig über viele Tausende Computer ausgeführt → so kann er mit der steigenden Anzahl an Webseiten skaliert werden. Grundsätzlich greift er alle paar Sekunden auf die gleiche Webseite zu. Damit die Bandbreite eines Servers nicht zu sehr beansprucht wird, kann bei Google eine Änderung der Crawling-Frequenz beantragt werden. (NP:Web21, vgl. developers.google.com 02.01.2021)

Weiters verwendet Google für andere Einsatzgebiete außer dem Indexieren ebenfalls Crawler → beispielsweise wird für das Anzeigen von relevanten Werbungen auf einer Webseite der AdSense-Bot oder AdsBot verwendet. Ein weiteres Beispiel wäre der Googlebot-Image, der für die Bildersuche auf Google zuständig ist. (NP:Web21, vgl. developers.google.com 02.01.2021)

3.8.2 Bingbot

Der Bingbot ist der von Microsoft entwickelte Crawler für deren eigene Suchmaschine Bing und ist ebenfalls dafür zuständig, dass einem User beim Verwenden von Bing relevante Ergebnisse angezeigt werden. Grundsätzlich ist die Funktionsweise des Bingbots der des Googlebots sehr ähnlich. (NP:Web41, vgl. avidera.com 09.02.2021)

2012 wurde die Microsoft Live Search durch Bing ersetzt und ab dann war der Bingbot für den Aufbau des Index zuständig. Außer Bing wird die Suchmaschine Yahoo auch von Microsoft betrieben → d.h. in beiden Suchmaschinen bekommt man für die gleiche Eingabe sehr ähnliche Ergebnisse. Yahoo benutzt aber neben dem Bingbot auch den Slurpbot um ihren Index zu ergänzen. (NP:Web41, vgl. advidera.com 09.02.2021)

Microsoft verwendet ebenfalls verschiedene Crawlerapplikationen für verschiedene Einsatzgebiete. Beispielsweise den “AdIdxBot”, welcher für das Platzieren der Bing Ads auf einer Webseite zuständig ist, oder den “BingPreview-Bot”, welcher Snapshots der Webseiten erstellt, die auch in den Bing Suchergebnissen erscheinen. (NP:Web42, vgl. bing.com 10.02.2021)

3.8.3 DuckDuckBot

Die Suchmaschine “DuckDuckGo” ist allgemein dafür bekannt, dass keine Informationen wie beispielsweise IP-Adressen gespeichert werden und die Nutzer nicht profiliert werden → wodurch jedem User dieselben Suchergebnisse angezeigt werden. Cookies werden nur verwendet, wenn sie wirklich benötigt werden. (NP:Web43, vgl. help.duckduckgo.com 10.02.2021)

Der DuckDuckBot ist für das Hinzufügen und Aktualisieren von Webseiten im DuckDuckGo-Index verantwortlich. Neben den Daten, die der eigene Bot liefert, werden auch Daten von Crawlern der anderen Suchmaschinen wie Bing, Yahoo oder Yandex für den Aufbau des Index verwendet. (NP:Web43, vgl. help.duckduckgo.com 10.02.2021)

3.9 Spider-Traps

Bei einer Spider-Trap handelt es sich um Mechanismen, die auf einer Webseite getroffen werden, um unerwünschte Crawler vom Indexieren der Seiten abzuhalten. Dazu macht man sich den Umstand zunutze, dass sich unerwünschte Crawler oft nicht an Vorgaben aus der robots.txt-Datei halten. (NP:Web26, seo-suedwest.de 06.01.2021)

Crawler, welche die eigentlich durch die robots.txt gesperrten Seiten besuchen, werden auf vom Server generierte unendliche Link-Strukturen umgeleitet. Diese Strukturen lassen den Crawler eine unendliche Anzahl an Links innerhalb der Domain crawlern, um ihn dort “gefangen” zu halten. (NP:Web22, vgl. techopedia.com 02.01.2021)

Crawler-Traps können vom Webseitenbetreiber aber auch unbeabsichtigt entstehen und Suchmaschinencrawler beim Indexieren der eigenen Webseite dadurch behindern. Die verschiedenen Arten von Traps und wie diese erzeugt werden, werden nachfolgend beschrieben.

Problem für das Crawl-Budget

Wie schon in Abschnitt 2.5 beschrieben, ist das Crawl-Budget das Ausmaß an Ressourcen, dass ein Suchmaschinencrawler zum Indexieren einer Webseite verwendet. Unbeabsichtigte Spider-Traps sind deswegen ein Problem für das Budget, da der Crawler unnötig Ressourcen für Seiten, welche eigentlich gar nicht existieren sollten, verschwendet. (NP:Web28, vgl. marketingtracer.com 30.01.2021)

Der Googlebot erkennt Spider-Traps oftmals, stoppt darauf das Crawling und ändert seine Frequenz, mit der er diese Seiten crawlt. Er kann aber selbst nicht verhindern, dass er wieder in dieselbe Trap gerät. Ressourcen des Crawl-Budget werden somit trotzdem weiterhin verschwendet. (NP:Web28, vgl. marketingtracer.com 30.01.2021)

Qualitätsprobleme

Die Seiten, die in einer solchen endlosen Rekursion vom Server erzeugt werden, sind meist alle ident. Das hat zur Folge, dass der Suchmaschinencrawler dies als Duplicate Content (siehe 2.6.1) wertet und die Webseite als qualitativ minderwertig einstuft, was wiederum das Crawl-Budget negativ beeinflusst. (NP:Web28, vgl. marketingtracer.com 30.01.2021)

Arten von Spider-Traps

- HTTPS/Subdomain Redirect Trap

Die HTTPS Redirect Trap ist die am häufigsten auftretende Falle. Der User wird hierbei beim Versuch sich mit der nicht sicheren Version (HTTP) der Webseite zu verbinden, auf die sichere HTTPS-Variante umgeleitet.

So landet man beim Verbinden mit `http://www.beispiel.at/seite` bei `https://www.beispiel.at` (NP:Web28, vgl. marketingtracer.com 30.01.2021)

Das Problem hierbei ist, dass `http://www.beispiel.at/seite` eigentlich zu `https://www.beispiel.at/seite` umgeleitet werden müsste. Suchmaschinencrawler sehen dies dann als falschen Redirect an, welcher als Soft-404-Fehler gilt. Crawl-Budget wird hier verschwendet, da die Robots versuchen, diese Seiten immer wieder zu crawlten, aber auf denselben Fehler stoßen.

- Filter Trap

Die Filter Trap oder auch Mix & Match Trap tritt auf, wenn auf einer Webseite Mechanismen zur Verfügung stehen, mit denen man bestimmte Objekte sortieren und filtern kann. Dies ist am häufigsten bei großen Online-Shops zu finden, wenn die Filter über normale crawlbare Links implementiert sind und nicht per JavaScript. Wenn die Filter untereinander kombinierbar sind, entstehen Unmengen an Links, die das Crawl-Budget unnötig aufbrauchen. (NP:Web29, vgl. portent.com 31.01.2021)

Jede Filtermöglichkeit erhöht die Anzahl der Links um ein Vielfaches von Zwei. Wenn zum Beispiel 40 verschiedene Filteroptionen verfügbar sind, gibt es 2^{40} Möglichkeiten diese Objekte zu filtern/sortieren → dementsprechend gibt es auch 2^{40} unterschiedliche URLs die eigentlich den selben Inhalt zeigen. (NP:Web29, vgl. portent.com 31.01.2021)

- Never-Ending URL Trap

Diese Trap entsteht, wenn ein eigentlich absoluter Link innerhalb der Seite falsch als relativer Link implementiert wird, und das führende "/" vergessen wird:

```
<a href="seite1">Seite 1</a>
```

Klicken auf diesen Link führt also zu:

```
https://www.beispiel.at/seite1  
https://www.beispiel.at/seite1/seite1  
https://www.beispiel.at/seite1/seite1/seite1  
...
```

So entstehen theoretisch unendliche Linkstrukturen, welche alle denselben Inhalt zeigen. (NP:Web28, vgl. marketingtracer.com 30.01.2021)

- Calendar Trap

Bei einer Calendar Trap ist, wie der Name schon sagt, ein Kalender-Objekt auf der Webseite das Problem. Der Kalender erstellt unendlich viele leere Seiten, auch für Daten die in der Zukunft liegen. Das ausgewählte Datum wird dabei in die URL aufgenommen (z.B.: `www.beispiel.at/seite?datum=2020-02`). Der Crawler wird so nie an ein Ende kommen und das Crawl-Budget für unnötige Seiten verschwenden. (NP:Web29, vgl. portent.com 31.01.2021)

- SessionID Trap

Eine SessionID wird dazu verwendet, um userspezifische Daten für die jeweilige Sitzung auf der Webseite (ohne Cookies zu verwenden) zu speichern. Die SessionID wird dann der URL als Parameter angehängt.

Beispielsweise:

```
https://www.beispiel.at/beispiel?session=A4C2EA2AB8CACEB9E  
https://www.beispiel.at/beispiel?jsessionid=A4C2EA2AB8CACEB9E  
https://www.beispiel.at/beispiel?sid=A4C2EA2AB8CACEB9E  
https://www.beispiel.at/beispiel?sessid=A4C2EA2AB8CACEB9E
```

Jedes mal wenn der Crawler die Seite besucht, bekommt er eine neue SessionID zugewiesen → also können für ein und dieselbe Seite viele verschiedene URLs entstehen. (NP:Web29, vgl. portent.com 31.01.2021)

3.10 Webcrawling Frameworks

Da das Thema Webcrawling und -scraping mit zunehmender Wichtigkeit von Daten immer bekannter wird, werden im Folgenden Frameworks vorgestellt, die eine gute Basis bieten, um selbst einen Crawler nach eigenen Bedürfnissen zu programmieren.

3.10.1 Scrapy

Scrapy is a fast high-level web crawling and web scraping framework, used to crawl websites and extract structured data from their pages. It can be used for a wide range of purposes, from data mining to monitoring and automated testing. (NP:Web30, docs.scrapy.org 02.02.2021)

Scrapy ist ein weit verbreitetes open-source Webcrawling Framework welches in Python geschrieben und unter der BSD⁵-Lizenz veröffentlicht wurde. Es ist auf Twisted aufgebaut, einem Framework zum Schreiben von asynchronen, ereignisgesteuerten Netzwerkprogrammen. Asynchron bedeutet, dass beim Senden von Anfragen an Webserver ein non-blocking mechanism verwendet wird und so mehrere Seiten parallel heruntergeladen werden können.

Aus diesem Grund kann Scrapy gut in Projekten mit einem hohen Volumen an zu crawlenden Domains verwendet werden. (NP:Web31, vgl. softkraft.co 02.02.2021)

⁵Berkeley Software Distribution

Zum Extrahieren von Daten aus dem HTML-Response verwendet Scrapy sogenannte "Selector", mit denen bestimmte Elemente mittels XPath oder CSS-Expressions selektiert werden können. Außerdem lässt sich Scrapy auch mit XML- und HTML-Parsern wie zum Beispiel BeautifulSoup oder lxml kombinieren, wobei Scrapy dann nur mehr für das Handlen der Responses und nicht für das Parsen der Seite zuständig ist. Zusätzlich bietet das Framework "Feed Exports" der gescrapeten Daten in Formaten wie beispielsweise JSON, JSON lines, CSV oder XML. (NP:Web30, vgl. docs.scrapy.org 02.02.2021)

Weiters kann mit Middlewares (→ einer Art Plugin) die Funktionalität bei der Verarbeitung von Responses zusätzlich erweitert werden. (NP:Web30, vgl. docs.scrapy.org 02.02.2021)

Im nachfolgenden Beispiel ist ein Crawler, welcher die Reviews zu einem bestimmten Amazonprodukt ausliest, implementiert. Basiert auf (NP:Web32, blog.datahut.co 02.02.2021):

```
1 import scrapy
2
3 class AmazonReviewsSpider(scrapy.Spider):
4     name = 'amazon_reviews'
5     allowed_domains = ['amazon.de']
6     myBaseUrl = "https://www.amazon.de/Apple-MMTN2ZM-EarPods-Lightning-
7         Connector/product-reviews/B01M1EEP0B/?reviewerType=all_reviews&
8             pageNumber="
9     start_urls=[]
10
11
12     def parse(self, response):
13         data = response.css('#cm_cr-review_list')
14         reviews = data.css('.review-rating')
15         comments = data.css('.review-text')
16
17         for n, review in enumerate(reviews):
18             print('stars: ' + ''.join(review.xpath('.//text()').extract()))
19             + ', comment: ' + ''.join(comments[n].xpath(".//text()").
20                 extract()) + '\n')
```

Listing 3.1: Scrapy Beispiel

```
1 stars: 5,0 von 5 Sternen, comment: Arrived quickly and work like a dream.  
      Have been using them for a couple of weeks. Very pleased. Genuine  
      Apple.  
2  
3 stars: 5,0 von 5 Sternen, comment: Exactly what I needed, would recommend  
4  
5 stars: 1,0 von 5 Sternen, comment: Owned these a month, within a couple  
      of weeks the play / pause button kept pressing itself, before the  
      whole button broke fully, and now the left earphone has no sound,  
      despite me only using the earphones normally and not treating them  
      badly at all.
```

Listing 3.2: Scrapy Beispieldausgabe

3.10.2 Jaunt

Jaunt ist eine Java-Library, welche Funktionen fürs Webscraping, für die allgemeine Web-Automatisierung (beispielsweise zum Testen von Funktionalitäten) und für JSON-Abfragen bereitstellt. (NP:Web44, vgl. jaunt-api.com 11.02.2021)

Durch einen integrierten headless⁶ Browser werden Webanfragen effizient ausgeführt. Jaunt bietet Methoden zum Senden von HTTP-Requests und zum Parsen der eingehenden, zugehörigen Responses in HTML-, XHTML-, XML- und JSON-Format. (NP:Web44, vgl. jaunt-api.com 11.02.2021)

Javascript wird von Jaunt zwar nicht unterstützt, dafür gibt es aber Jaantium → eine Library, die auf Jaunt und Selenium basiert und beide Funktionalitäten miteinander vereint. So können auch dynamisch erzeugte Inhalte auf Webseiten geparsed werden. (NP:Web45, vgl. jaantium.com 11.02.2021)

Mit Jaunt ist es auch möglich Formulare auf Webseiten per Label auszufüllen, Dateien herunter- und hochzuladen, sowie komplett Webseiten inklusive der dargestellten Bilder zu speichern. (NP:Web44, vgl. jaunt-api.com 11.02.2021)

Im nachfolgenden Codebeispiel ist ein einfacher Scraper implementiert, welcher eine bestimmte URL, in diesem Fall <https://jaunt-api.com/jaunt-tutorial.htm>, bekommt und anschließend den Titel und den Inhalt der Seite ausgibt.

⁶besitzt keine Benutzeroberfläche

```

1 import com.jaunt.*;
2
3 public class JauntBeispiel {
4     public static void main(String[] args) {
5         try{
6             UserAgent userAgent = new UserAgent();
7
8             userAgent.visit("https://jaunt-api.com/jaunt-tutorial.htm");
9             String title = userAgent.doc.findFirst("<title>").getChildText();
10            String body = userAgent.doc.findFirst("<body>").getTextContent();
11
12            System.out.println("\nTitle: " + title);
13            System.out.println("\nContent: " + body);
14
15        }
16        catch(JauntException e){
17            System.err.println(e);
18        }
19    }
20 }
```

Listing 3.3: Jaunt Beispiel

```

1 Title: Jaunt Webscraping Tutorial
2
3 Content: Jaunt Webscraping Tutorial - Quickstart
4
5 The Jaunt package contains the class UserAgent, which represents a
   headless browser. When the UserAgent loads an HTML or XML page, it
   creates a Document object.
6 ...
```

Listing 3.4: Jaunt Beispieldausgabe

3.10.3 Goutte

Goutte ist eine Scraping-, sowie Webcrawling-Library für PHP. Sie basiert auf Komponenten des Symfony Frameworks (→ ein Framework, welches Funktionen zum Erstellen von Webseiten und Webapplikationen bereitstellt), genauer auf BrowserKit, CssSelector, DomCrawler und HttpClient. (NP:Web35, vgl. [github.com 05.02.2021](https://github.com/05.02.2021))

BrowserKit simuliert das Verhalten eines Webbrowsers und bietet Funktionen zum Senden von Requests, Klicken von Links oder dem Absenden von Formularen. (NP:Web36, vgl. symfony.com 05.02.2021)

DomCrawler bietet Methoden zur DOM⁷-Navigation, sowie Funktionalität zum Selektieren und Filtern von XML- oder HTML-Elementen innerhalb eines Response. (NP:Web37, vgl. symfony.com 05.02.2021)

Die CssSelector Komponente wandelt CSS-Selektoren in XPath-Expressions um, so dass Elemente per XPath selektiert werden können. (NP:Web38, vgl. symfony.com 05.02.2021)

Wie es der Name schon vermuten lässt, ist der HttpClient dafür zuständig, HTTP-Requests zu erstellen, zu senden und zu empfangen. (NP:Web39, vgl. symfony.com 05.02.2021)

Im Folgenden ist eine simple Goutte-Applikation, die auf der Wikipedia-Startseite nach einem bestimmten Beitrag sucht und dann alle p-Tags auf dieser Seite in der Kommandozeile ausgibt, implementiert. Basiert auf (NP:Web40, medium.com 06.02.2021):

```

1 <?php
2 require 'vendor/autoload.php';
3 use Goutte\Client;
4
5 $client = new Client();
6 $crawler = $client->request('GET', 'https://www.wikipedia.org/');
7 $form = $crawler->filter('#search-form')
8     ->form(['search' => 'web crawler']);
9
10 $crawler = $client->submit($form);
11
12 echo $crawler->filter('.mw-parser-output p')->each(function ($node) {
13     print $node->text()."\n";
14 });
15 ?>
```

Listing 3.5: Goutte Beispiel

```

1 C:\Users\Nicolas\Desktop\Schulordner 5CHIF\Diplomarbeit\GoutteCrawler>php
2 Main.php
3
4 A Web crawler, sometimes called a spider or spiderbot and often shortened
5 to crawler, is an Internet bot that systematically browses the World
Wide Web, typically for the purpose of Web indexing (web spidering).
Web search engines and some other websites use Web crawling or spidering
software to update their web content or indices of other sites' web
content. Web crawlers copy pages for processing by a search engine,
which indexes the downloaded pages so that users can search more
efficiently.
5 ...
```

Listing 3.6: Goutte Beispieldausgabe

⁷Document Object Model

Kapitel 4

Implementation eines Webcrawlers im Projekt AI Börse

4.1 Problemstellung

Im Zuge der Arbeit war es ein Teilziel, einen Crawler zu programmieren, welcher die eigentlichen Daten für die spätere Bewertung der verschiedenen Börsennewsseiten bereitstellen soll. Dazu sollte das Programm an den Wochentagen, an denen die Börse geöffnet hat, jeweils stündlich innerhalb der Öffnungszeiten ausgeführt werden und demnach immer aktuelle Texte der Anbieter erfassen.

Im Vorhinein ist es wichtig, die zu verwendenden Quellen, von denen die Testdaten bezogen werden sollen, festzulegen. Dies erfordert eine Evaluierung der aktuellen Marktsituation an Webseiten, welche Beiträge und Artikel zum DAX veröffentlichen. Viele Anbieter veröffentlichen zwar News, jedoch können diese in der Arbeit nicht verwendet werden, da in diesen Berichten meist nur über den Momentanzustand des DAX geschrieben wird. Für die Umsetzung des Projekts werden allerdings Prognosen über die zukünftige Entwicklung des Index benötigt.

Weiters spielt die Verständlichkeit der Texte auch eine entscheidende Rolle für die Auswertung. Unverständlich und verwirrend geschriebene Texte sind schwerer zu analysieren als Texte, die die eigentlich wichtige Information kurz und knapp darstellen. Alle verwendeten Quellen sind in folgender Tabelle mit kurzer Beschreibung angeführt.

Quelle (URL)	Bemerkung
https://www.onvista.de/news/boerse-daily/	Einige Beiträge sind eher unverständlich, der Großteil ist aber verständlich
https://www.boerse-daily.de/news-analysen-insight	Leicht und einfach geschriebene Artikel, klar ersichtlich was prognostiziert wird
https://finanzmarktwelt.de/dax/	Beiträge mit vielen wichtigen Kennzahlen, nicht verwirrend und einfach zu verstehen
https://de.investing.com/indices/germany-30-opinion	Analysen und Meinungen von Experten, einfache und verständliche Prognosen
https://www.boerse-online.de/maerkte/dax-chartanalyse	Tägliche DAX-Chartanalysen, relevante Unterstützungen und Widerstände werden angegeben
https://www.dailyfx.com/deutsch/finanznachrichten	Tägliche charttechnische Prognosen, beinhalten wichtige Kennzahlen
https://admiralmarkets.com/de/analysen/dax30-tages-updates	Sehr lange Berichte, es werden viele mögliche Szenarien, wie sich der DAX entwickeln könnte beschrieben, dennoch einfach zu verstehen
https://www.godmode-trader.de/indizes/dax-performance-index-kurs,133962	Kurze tägliche Prognosen von Experten, Auskunft über relevante Kennzahlen
https://app.libertex.com/products/indexes/FDAX/	Liefert mehrmals täglich Prognosen, wichtige Daten werden formatiert dargestellt

Tabelle 4.1: Liste der verwendeten Quellen

4.2 Spezifikation

Als Basis für die Programme wurde das Webcrawling Framework Scrapy (beschrieben in 3.10.1) und die Programmiersprache Python gewählt. Scrapy stellt für die Anwendung, die beste und einfachste Webcrawling- und Scraping-Funktionalität zur Verfügung. Weiters ist es sehr simpel ein Pythonprogramm per Crontab auf einem Linux-Server automatisiert ausführen zu lassen.

Für die Webseiten, welche keine dynamischen Inhalte darstellen, wird ein Crawler (Linkcrawler) entwickelt, welcher die URLs zu neu veröffentlichten Artikeln ausliest und diese weiter an einen Scraper (Contentcrawler) liefert. Der Scraper hat dann die Aufgabe, die benötigten Informationen zu extrahieren und zu filtern und anschließend in der Datenbank abzuspeichern.

Damit dynamische Inhalte auf den Webseiten geladen werden können, benötigt es zusätzlich das Framework Selenium, da Scrapy alleine nur statische Inhalte anzeigen kann. Dadurch, dass ein Crawlvorgang, gestützt von Selenium, in der Regel um einiges länger dauert, wird ein eigener Crawler (Libertexcrawler) entwickelt, welcher für die Webseiten zuständig ist, die dynamische Inhalte verwenden. Dies ist nur bei "https://app.libertex.com/" der Fall.

Beide Crawler werden täglich zwischen den Handelszeiten (9 bis 18 Uhr) an der deutschen Börse automatisiert ausgeführt und für jede festgelegte Quelle werden nur neu veröffentlichte Prognosen ausgelesen und verwertet.

4.3 Arbeitsprozess

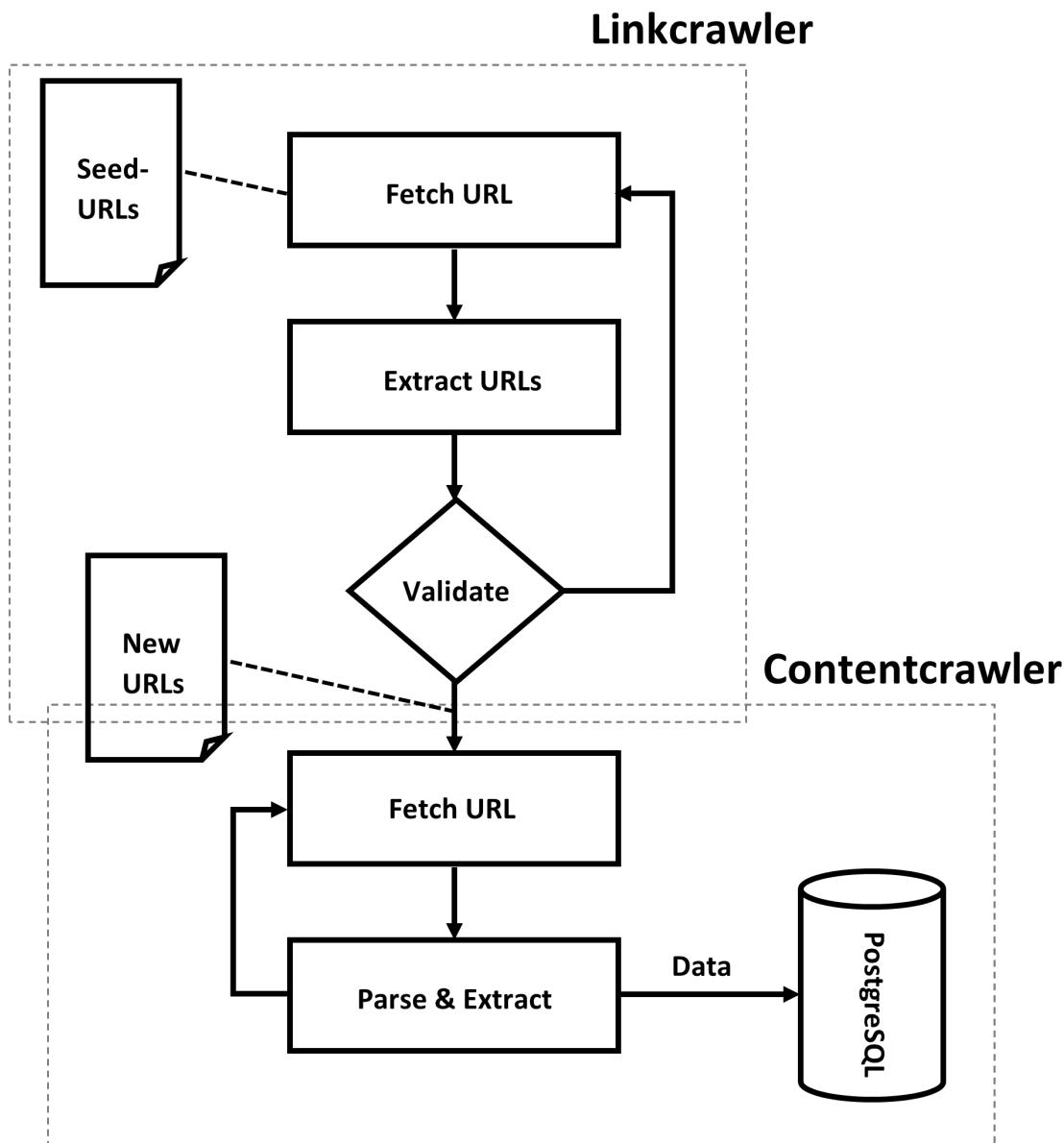


Abbildung 4.1: Arbeitsprozess Link- & Contentcrawler

Der Linkcrawler ist dafür zuständig, aus den vorgegebenen URLs (Seed-URLs) alle gefundenen Links zu extrahieren und diese zu validieren, sprich, zu prüfen, ob der Link relevant ist und zu einem neu veröffentlichten Artikel führt. Trifft dies zu, dann werden die neuen URLs an den Contentcrawler weitergeleitet, welcher aus diesen Links den benötigten Inhalt filtert und in der Datenbank abspeichert.

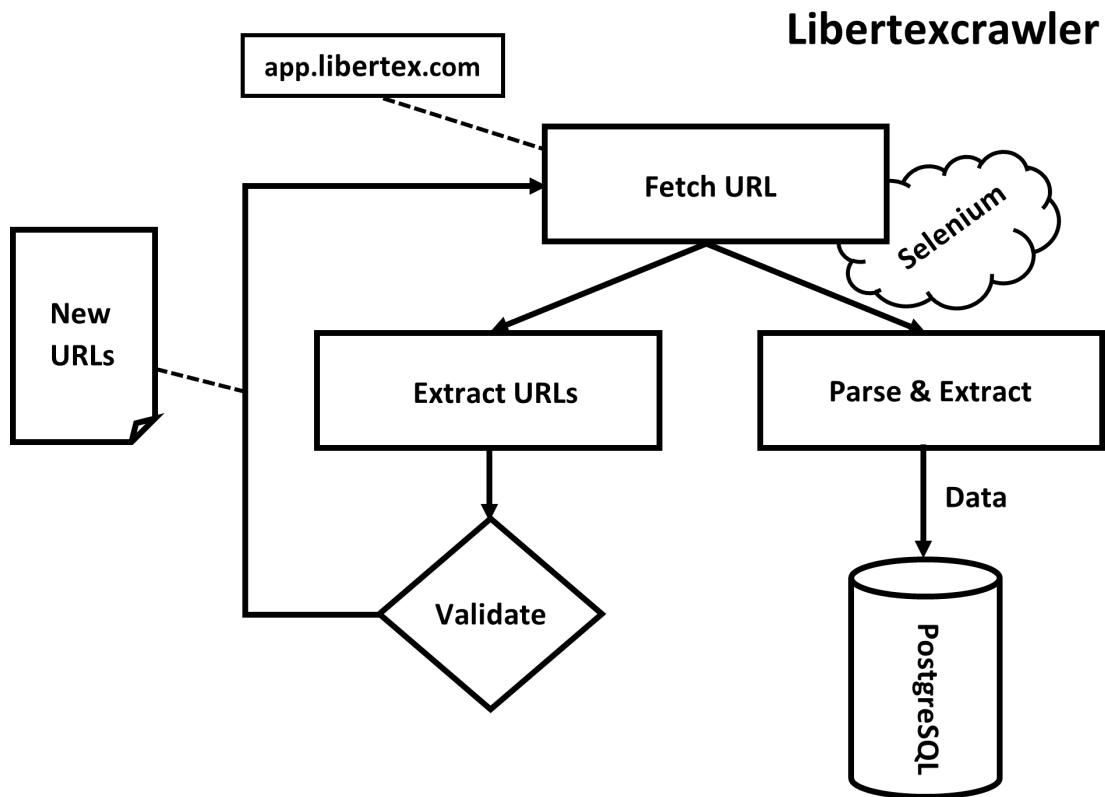


Abbildung 4.2: Arbeitsprozess Libertexcrawler

Im Gegensatz zur Link- & Contentcrawler-Architektur ist der Libertexcrawler eigens für das Finden neuer Links und das Extrahieren der relevanten Daten zuständig. Dabei startet er bei einer vorhin festgelegten URL, an der alle Artikel gefunden werden können. Mit dem Framework Selenium ist es möglich einen normalen User im Webbrowser zu simulieren, so können auch JavaScript-Inhalte geladen werden. Der Crawler extrahiert im ersten Durchlauf alle neuen Links und filtert anschließend relevanten Text aus den gefundenen Beiträgen und speichert diesen in der Datenbank.

Die Implementation der Crawler wird im folgenden Abschnitt 4.4 genauer beschrieben.

4.4 Implementation

In den nachstehenden Abschnitten ist der Code von Link- & Contentcrawler, sowie dem Libertexcrawler aufgeteilt und beschrieben. Nicht relevante Codeteile wurden dabei gekürzt/ausgestrichen. Der vollständige Quellcode der Programme ist auf dem Datenträger unter X:\Programme\Crawler zu finden.

4.4.1 Linkcrawler

```

1  class Crawler(CrawlSpider):
2      name = "Linkcrawler"
3
4      start_urls = [
5          "https://www.onvista.de/news/",
6          ...
7      ]
8
9      custom_settings = {
10         ...
11         'DEPTH_LIMIT': 1,
12         'CONCURRENT_REQUESTS_PER_DOMAIN': 32,
13         'DUPEFILTER_CLASS': 'scrapy.dupefilters.RFPDupeFilter',
14     }
15
16     rules = (
17         Rule(LinkExtractor(deny=(r'\.pdf', r'\?pdf')), callback='parse'),
18     )
19
20     def parse(self, response):
21         url = response.url
22         x = re.search(regex, url)
23
24         regexList = [boerseO_regex, boerseD_regex, godmode_regex1,
25                     godmode_regex2, onvista_regex1, onvista_regex2, investing_regex,
26                     dailyfx_regex, finanzmarktwelt_regex]
27
28         if x is None:
29             if any(regex.match(url) for regex in regexList):
30                 linkPresent = get_FreitextByLink(url)
31                 if not linkPresent:
32                     item = LinkItem(link_url=url)
33                     yield item

```

Listing 4.1: Linkcrawler

Der Linkcrawler bekommt die im "start_urls"-Array angegebenen URLs als Seed-Set übergeben und nimmt sich von dort die Links, bei denen er beginnt neue Artikel zu suchen. Die Klasse erbt von scrapy.CrawlSpider, damit keine eigene Funktionalität implementiert werden muss, welche bewirkt, dass der Crawler selbstständig den gefundenen Links folgt; oder auch nicht folgt. Weiters können dadurch auch Regeln definiert werden, die das Programm beim Folgen der Links beachten soll. In diesem Fall soll der Crawler keinen PDFs folgen. Dies bewirkt die Rule mit dem Schlüsselwort "deny" und der angegebenen Regex in Codezeile 17. Die Methode, die im "callback"-Attribut angegeben wird, wird für jeden Response, den der Crawler zurückbekommt, aufgerufen.

Mithilfe der "custom_settings" können bestimmte Einstellungen festgelegt werden, die das Programm beim Crawlen befolgen soll. Mit der "DEPTH_LIMIT" wird eingestellt, wie tief der Crawler in der Webseitenstruktur suchen soll, mit "CONCURRENT_REQUESTS_PER_DOMAIN" kann festgelegt werden, wie viele Anfragen gleichzeitig an den selben Webserver vom Crawler gestellt werden können. In diesem Fall ist die Tiefe auf 1 festgelegt, da der Crawler jeweils nur bei den angegeben Start-URLs nach den Artikeln suchen soll und nicht wirklich tief in die Linkstruktur der Webseite eindringen soll. Der Dupefilter (Zeile 13) verhindert, dass das Programm in eine unendliche Rekursion läuft → beispielsweise 2 Seiten verlinken einander gegenseitig.

In der Parse-Methode werden die gefundenen Links weiter mittels Regex eingegrenzt, sodass nur die URLs weiterverarbeitet werden, hinter denen sich auch wirklich ein relevanter Artikel befindet.

Zuletzt wird mit der "get_FreitextByLink"-Methode mit der Datenbank verglichen, ob der jeweilige ausgelesene Link nicht schon gespeichert ist, also dementsprechend nicht neu veröffentlicht wurde. Weiters wird ein neues "LinkItem" (erbt von scrapy.Item) mit dem neuen Link erstellt. Beim Aufruf des Programmes über die Kommandozeile wird dieses dann mithilfe der Scrapy Pipeline einem JSON-File hinzugefügt, welches nach einem erfolgreichen Durchlauf alle Links zu den neuen Artikeln in JSON-Format beinhaltet.

Der Aufruf über die Kommandozeile könnte beispielsweise so aussehen; der Dateiname spielt dabei keine Rolle:

```
scrapy crawl Linkcrawler -o outputLinks.json
```

```
1 [
2     {"link_url": "https://finanzmarktwelt.de/dax-daily-die-devise-an-den-",
3      "aktienmaerkten-lautet-to-the-moon-190317/"},
4     {"link_url": "https://www.boerse-online.de/nachrichten/ressort/maerkte/",
5      "dax-chartanalyse-anstieg-wird-sich-verlangsamten-1030044819"},
6     {"link_url": "https://www.boerse-daily.de/boersen-nachrichten/insight-dax-",
7      "ruhiger-handel-vor-us-leitzinsentscheid-34490"},
8     {"link_url": "https://www.godmode-trader.de/analyse/dax-tagesausblick-",
9      "weiterer-sprunghafter-kursanstieg,9290831"}
10    ...
11 ]
```

Listing 4.2: outputLinks.json

4.4.2 Contentcrawler

```

1  class Crawler(scrapy.Spider):
2      name = "Contentcrawler"
3
4      def make_requests_from_url(self, url):
5          return Request(url, dont_filter=True, meta = {
6              'dont_redirect': True,
7              'handle_httpstatus_list': [301, 302]
8          })
9
10     def start_requests(self):
11         f = open(self.inputfile, "r", encoding='utf-8')
12         data = json.loads(f.read())
13         f.close()
14
15         url_list = []
16
17         for entry in data:
18             url_list.append(entry['link_url'])
19
20         for url in self.static_linklist:
21             url_list.append(url);
22
23         for url in url_list:
24             yield self.make_requests_from_url(url)

```

Listing 4.3: Contentcrawler

Der Contentcrawler erbt von der Klasse scrapy.Spider, welche keine zusätzlichen Funktionen, außer der “start_requests”- und der “parse”-Methode, bereitstellt. In diesem Fall wird beim Starten des Programms das zuvor durch den Linkcrawler erstellte JSON-File, welches die neuen Links zu den Artikeln beinhaltet, per Kommandozeile übergeben und in der “start_requests”-Funktion ausgelesen. Mithilfe der “make_requests_from_url”, die dahingehend überschrieben wurde, dass die Requests nicht redirected werden, wird in einer Schleife an jeden verfügbaren Link eine Anfrage geschickt.

```

25     def parse(self, response):
26         text = []
27         content = response.xpath('...')
28
29         for line in content.xpath('.//p//text()').extract():
30             text.append(processLine(line))
31
32         filteredInput = []
33
34         for line in text:
35             if any(x in line for x in keywords):
36                 filteredInput.append(endSentence(line))

```

```

37     timestamp = datetime.datetime.now()
38     formattedTimestamp = timestamp.strftime('%d-%m-%Y; %H:%M:%S')
39     splitDate = formattedTimestamp.split(';')
40
41     Beitrag = ContentItem(freitext=filteredInput, url=response.request.
42         url, datum=splitDate[0], Uhrzeit=splitDate[1])
43     add_Freitext(Beitrag)

```

Listing 4.4: Contentcrawler

Für jede Antwort, die nach einem Request vom Webserver zurückkommt, wird die “parse”-Methode aufgerufen. Beim Parsen wird aus dem Response per XPath-Expression das jeweilige Element, welches den benötigten Inhalt enthält, selektiert. In den meisten Fällen ist dies ein div- oder article-Element. Anschließend wird aus diesem Selector-Objekt der Textinhalt aller p-Tags extrahiert, formatiert und anschließend einem Array angefügt. Daraufhin wird jedes Element des zuvor erstellten Arrays mit einem Set an im Vorhinein festgelegten Keywords abgeglichen. Wenn ein Keyword im Text vorkommt, dann wird dieser als relevant angesehen, einheitlich formatiert und einem weiteren Array, welches den gefilterten Content beinhaltet, angefügt.

Zuletzt wird ein neues “ContentItem” erstellt, welches von scrapy.Item erbt und den gefilterten Text, die zugehörige URL und das Datum mit Uhrzeit zum Zeitpunkt des Crawlingvorganges speichert. Mit der “add_Freitext”-Methode werden die gesammelten Daten in der Datenbank abgespeichert.

Der Aufruf über die Kommandozeile sieht dann folgenderweise aus → der Dateiname, der für den Parameter “inputfile” angegeben wird, muss mit dem Namen übereinstimmen, welcher als Outputfile bei der Ausführung des Linkcrawlers gewählt wurde:

```
scrapy crawl Contentcrawler -a inputfile=outputLinks.json
```

```

1 ...
2 2021-03-20 15:19:53 [scrapy.core.engine] DEBUG: Crawled (200) <GET https://admiralmarkets.com/de/analysen/dax30-tages-updates> (referer: None)
3 ['Der Dax ging gestern morgen bei 14.725 Punkten in den vorbörslichen Handel. ...', 'Kann sich der Dax über der 14.680 Punkte Marke halten, so könnte es weiter aufwärts an unsere nächsten Anlaufziele bei 14.689/91, bei 14.702/04, bei 14.711/13 und dann bei 14.723/25 Punkten gehen. Kommt es hier zu keinen Rücksetzern, so könnte sich die Aufwärtsbewegung weiter fortsetzen.', 'Rutscht der Dax unter die 14.680 Punkte, so könnte er zunächst unsere nächsten Anlaufziele bei 14.669/67, bei 14.651/49 und dann bei 14.636/34 Punkten erreichen. Kommt es hier zu keinen Erholungen, so könnten sich die Abgaben weiter fortsetzen.', '14.763 Punkte bis 14.635 Punkte ist die heute von uns erwartete Tagesrange. ']
4 ...

```

Listing 4.5: Ausgabe Contentcrawler

4.4.3 Libertexcrawler

```

1  class Crawler(scrapy.Spider):
2      name = "Libertexcrawler"
3
4      allowed_domains = [
5          "app.libertex.com",
6      ]
7
8      start_urls = [
9          "https://app.libertex.com/products/indexes/FDAX/",
10     ]
11
12     def parse(self, response):
13         regexp = re.compile(r'#modal_news_.*_FDAX')
14
15         options = webdriver.ChromeOptions()
16         options.add_argument("--headless")
17         capabilities = options.to_capabilities()
18         driver = webdriver.Chrome(desired_capabilities=capabilities)
19
20         driver.get(response.request.url)
21         sel = Selector(text=driver.page_source)
22
23         if "https://app.libertex.com/products/indexes/FDAX/" == response.
24             request.url:
25             links = sel.xpath("//a/@href").extract()
26
27             for l in links:
28                 if regexp.search(l):
29                     newUrl = response.request.url + l
30
31                     linkPresent = get_FreitextByLink(newUrl)
32                     if not linkPresent:
33                         yield self.make_requests_from_url(newUrl)

```

Listing 4.6: Libertexcrawler

Der Libertexcrawler umfasst einerseits das Erfassen neuer Links, sowie das Extrahieren der Daten aus den gefundenen Links. Die Klasse erbt wie auch der Contentcrawler von scrapy.Spider und bekommt somit auch keine spezielle Funktion mitgeliefert.

Das Programm bekommt über das “start_urls”-Array den Link übergeben, wo alle neuen Artikel, die erfasst werden sollen, veröffentlicht werden. Bei Programmstart wird ein HTTP-Request an diese URL gestellt und für den Response, welcher vom Webserver zurückkommt, wird die “parse”-Methode aufgerufen. Da “https://app.libertex.com” JavaScript-Inhalte verwendet, ist es erforderlich, die Webseite mit Selenium zu laden. In Codezeile 15 bis 18 wird der Browser, in diesem Fall Chrome und die zu benutzenden Einstellungen festgelegt. Das Argument “--headless” bewirkt, dass die Seite ohne grafi-

sche Benutzeroberfläche geladen wird. In Zeile 20 wird dann mittels “get”-Methode des webdriver-Objekts die Anfrage gestellt. Der Inhalt der Seite kann weiters über einen Selector ausgelesen werden.

Im ersten Durchlauf des Crawlers, also in dem Durchlauf in dem er neue Artikel finden soll, werden über das Selector-Objekt mittels XPath-Expression alle a-Tags auf der Seite extrahiert. Für jeden dieser gefundenen Links wird überprüft, ob dieser die in Codezeile 13 definierte Regex matched. Stimmt der Link mit dem Ausdruck überein und besteht in der Datenbank nicht schon ein Eintrag mit dem gefundenen Link, dann wurde ein neuer Artikel gefunden. Für diese Links wird dann die “make_requests_from_url”-Methode aufgerufen, welche bewirkt, dass neue Anfragen an diese URLs gestellt werden.

```

33     if regexp.search(response.request.url):
34         article = sel.xpath('//div[@class="article"]')
35         body = article.xpath('.//div[@class="article-body"]//text()').
36             extract()
36         heading = article.xpath('.//h2[@class="section-title"]//text()').
37             extract()
37         articleDate = article.xpath('.//span[@class="date"]//text()').
38             extract()
38
39         if heading and articleDate:
40             if ("Dax im Tagesverlauf" in heading[0] or "Dax (Eurex) (H1) im
41                 Tagesverlauf" in heading[0]) and date in articleDate[0]:
41                 for x in range(8,18):
42                     if str(x) in hour:
43                         input = []
44
45                         for line in body:
46                             line = line.replace("\n", " ")
47                             line = line.replace("*", "")
48                             line = line.strip()
49
50                         if line:
51                             if any(x in line for x in keywords) or any(char.isdigit
51                                 () for char in line):
52                                 input.append(endSentence(line))
52
53
54             timestamp = datetime.datetime.now()
55             formattedTimestamp = timestamp.strftime('%d-%m-%Y;%H:%M:%S'
55                 )
56             splitDate = formattedTimestamp.split(';')
57
58             beitrag = ContentItem(freitext=input, url=response.request.
58                 url, datum=splitDate[0], uhrzeit=splitDate[1])
59
59             add_Freitext(beitrag)
60             break
61

```

Listing 4.7: Libertexcrawler

Für die Responses, die das Programm von den neuen Artikeln zurückbekommt, wird nun, wieder mit XPath-Expressions, das gelieferte HTML-Dokument geparsed. Über ein div-Element, welches den gesamten Inhalt des Artikels umfasst, wird das Heading, der Body und das Artikeldatum extrahiert.

Nur Artikel welche “Dax im Tagesverlauf” oder “Dax (Eurex) (H1) im Tagesverlauf” im Titel beinhalten, sind für die Auswertung interessant. In den anderen Beiträgen werden keine richtigen Prognosen getroffen. Zudem muss das Artikeldatum mit dem Datum zum Zeitpunkt der Ausführung übereinstimmen und der Artikel während der Börsenöffnungszeiten veröffentlicht worden sein, somit werden nur aktuelle Texte weiterverarbeitet.

Nach der Validierung wird der Text aus dem Body zeilenweise gelesen und die jeweilige Zeile formatiert. Der Inhalt der Zeile wird auch hier wieder darauf überprüft, ob bestimmte Keywords im Text vorkommen. Darüber hinaus wird auch überprüft ob der Text eine Zahl enthält. Wenn eine dieser beiden Bedingungen zutrifft, wird der Text einem Array angehängt.

Danach wird ein neues “ContentItem” erstellt, dass den gefilterten Text, die Response-URL und das aktuelle Datum mit Uhrzeit speichert. Zuletzt wird der neu erfasste Freitext in der Datenbank abgespeichert.

Der Crawler kann über die Kommandozeile mit nachstehendem Aufruf ausgeführt werden; dabei werden keine speziellen Parameter benötigt:

```
scrapy crawl Libertexcrawler
```

```
1 ...
2 2021-03-01 15:10:41 [scrapy.core.engine] DEBUG: Crawled (200) <GET https://app.libertex.com/products/indexes/FDAX/#modal_news_5711895_FDAX> (referer: https://app.libertex.com/products/indexes/FDAX/)
3 2021-03-01 15:10:42 [selenium.webdriver.remote.remote_connection] DEBUG: Finished Request http://127.0.0.1:52305/session/43
4 c18f59a13ba90527cd707167860bfe/url {"url": "https://app.libertex.com/products/indexes/FDAX/#modal_news_5711895_FDAX"}
5 ...
6 ['Kaufposition über 13830,00 mit Kurszielen von 14010,00 & 14070,00. ', '
Pivot-Punkt: 13830,00. ', 'Unsere Meinung: Kaufposition über 13830,00
mit Kurszielen von 14010,00 & 14070,00. ', '...']
```

Listing 4.8: Ausgabe Libertexcrawler

4.5 Automatisierung

4.5.1 Crontab

Damit die Daten täglich automatisiert erfasst werden, benötigt es ein Shell-Script welches mittels crontab auf einem Linux-Server ausgeführt wird.

In unten angeführtem Skript, wird der Linkcrawler gestartet um neue Links zu Artikeln zu finden. Nach dessen Ausführung wird der Contentcrawler mit dem zuvor erstellen File als "inputfile" ausgeführt. Anschließend daran, wird der Crawlvorgang des Libertex-crawlers gestartet.

```
1 #!/bin/sh
2 cd Programme/Crawler/LinkContentCrawler/crawler
3 scrapy crawl Linkcrawler -o outputLinks.json
4
5 scrapy crawl Contentcrawler -a inputfile=outputLinks.json
6 rm outputLinks.json
7
8 cd Programme/Crawler/LibertexCrawler/crawler
9 scrapy crawl Libertexcrawler
```

Listing 4.9: CrawlingScript.sh

```
1 0 9-18 * * 1-5 /home/schueler/CrawlingScript.sh
```

Automatisch ausgeführt wird das Shell-Script durch einen Eintrag in der crontab-Tabelle. Oben angeführter crontab-Eintrag bewirkt, dass das Shell-Script automatisch zu jeder vollen Stunde (0) zwischen 9 und 18 Uhr (9-18), an jedem Tag (*), jedem Monat (*) und von Montag bis Freitag (1-5) vom System ausgeführt wird.

4.6 Fazit

Die Wahl des Webcrawling-Frameworks Scrapy hat sich im Laufe der Arbeit als sinnvoll erwiesen, da dies eines der meist genutzten Frameworks für diesen Anwendungsfall ist und dadurch auch dementsprechend viele Best-Practices im Internet zu finden sind.

Durch die vorgefertigte Funktionalität musste nur weiters implementiert werden, wie neue Links gefunden und der gewünschte Text extrahiert wird.

Im Zuge der Implementation des Linkcrawlers stellte sich allerdings die Herausforderung, dass der Crawler zu tief in der Linkstruktur der Webseiten gesucht hat und auf irrelevante Webseiten umgeleitet wurde. Dadurch konnte das Programm nur schwer neue Artikel finden. Das Problem wurde durch die Erstellung einer Regex gelöst, die verhindert, dass der Crawler Links folgt, denen er eigentlich nicht folgen sollte.

Wie auch im Abschnitt 4.2 näher beschrieben, musste eine Lösung gefunden werden, die es ermöglicht, JavaScript-Inhalte auf Webseiten zu laden, da das Framework Scrapy dies alleine nicht kann. Dazu wurde das Framework Selenium gewählt, welches einen normalen User im Webbrower simulieren kann.

Durch das Lösen dieser Aufgaben, verlief die weitere Implementierung problemlos. Darüber hinaus ist es auch möglich, die Programme in Zukunft mithilfe von Scrapy Middlewares, um zusätzliche Funktionalitäten zu erweitern.

Die Automatisierung des Crawlingvorganges erfolgte durch die automatische Ausführung der Programme durch einen crontab-Eintrag auf einem Linux-Server. Dies war dabei die einfachste Lösung diese Aufgabe zu erfüllen.

Das Ziel, Texte von Webseiten, welche Prognosen zur Entwicklung des DAX veröffentlichen, automatisch auszulesen, wurde somit vollständig erfüllt.

Kapitel 5

Clustersysteme

5.1 Was ist ein Clustersystem

Wenn man mit *Big Data* arbeitet, benötigt man sehr hohe Rechenleistungen. Zu Beginn war es noch üblich, diese Rechenkapazität nur über ein Gerät zu erreichen, einen sogenannten Supercomputer, das führte jedoch schnell zu sehr hohen Kosten. Daher schreibt die derzeitige Norm für die Verarbeitung von großen Datenmengen vor, Clustersysteme heranzuziehen.

Ein Clustersystem ist ein Verbund, bestehend aus mehreren einzelnen Rechnern, die miteinander kommunizieren, und daher von außen als ein System wahrgenommen werden können. Ein einzelner Rechner aus so einem Verbund wird Node (de. Knoten) genannt und dazu verwendet, kleine Teilaufgaben zu erledigen. Diese fließen dann in ein Gesamtergebnis zusammen. Damit das funktioniert, müssen sich die Knoten andauernd untereinander austauschen.

1967 war erstmals die Rede von Cluster Computing, in einem Paper zu dem Thema '*Parallel Processing*' von Gene Amdahl, ein Computerarchitekt bei IBM. In dem Paper wurde die Basis für Multiprozessoren sowie Clustersysteme gebildet. Auch das Amdahl's Law 5.3.1 entstand aus diesem Paper heraus.

1969 wurde das ARPANET Projekt gegründet, welches auf ein Paketvermittelndes Netzwerk aufsetzt. Bei diesem Projekt wurden vier Computercenter miteinander verbunden und jedes dieser Center war ähnlich wie ein Computercluster aufgebaut. Aus dem ARPNET hat sich später das Internet entwickelt, weswegen es auch als die 'Mutter' der Computer Cluster gilt.

Mehr Informationen zu Cluster Computing werden auf folgender Seite gegeben: (LR:Web03, vgl. cloudflight.io 31.10.2020)

5.2 Arten von Clustersystemen

Es gibt vier unterschiedliche Arten von Clustern, die zwar alle ähnlich funktionieren, doch ein unterschiedliches Ziel verfolgen:

- **High Availability-Cluster (HA-Cluster)**

High Availability-Cluster, oder Hochverfügbarkeit Cluster, haben das Ziel, für eine bessere Ausfallsicherheit zu sorgen. Das gelingt ihnen, indem sie einen Single-Point-of-Failure – der Ausfall von einem Teil führt zum Stillstand des gesamten System – verhindern. Wenn nämlich ein Knoten in dem Cluster ausfällt, werden dessen Aufgaben von einem anderen übernommen, daher reichen bereits zwei Knoten aus, um einen solchen Cluster aufzubauen.

Bei dem Betrieb von einem HA-Cluster gibt es zwei Möglichkeiten:

- "active/active"

Bei dieser Variante verarbeiten beide Knoten während dem normalen Betrieb die selbe Art von Aufgaben. Sollte einer der beiden Knoten ausfallen, übernimmt der andere dessen Aufgaben. Diese Variante nennt man auch Takeover.

- "active/passive"

Bei dieser Variante verarbeitet nur ein Knoten die Aufgaben im normalen Betrieb und ein anderer ist in Standby. Der zweite Knoten befindet sich solange in Ruhephase, bis es zu einem Ausfall des ersten Knoten kommt. Dieses Verfahren nennt man Failover.

Anwendungsbereich solcher Cluster ist überall, wo die Ausfallzeit im Jahr nicht mehr als paar Minuten betragen darf.

Häufig werden die unterschiedlichen Knoten auch noch räumlich getrennt, damit diese auch katastrophensicher sind. Das nennt man dann "Stretched Cluster".

- **High Performance Computing-Cluster (HPC-Cluster)**

High Performance Computing-Cluster, oder auch häufig Superrechner genannt, werden meist dazu verwendet, um anspruchsvolle Rechenaufgaben auf mehreren Datensätzen durchzuführen. Dabei wird die Aufgabe meist in mehrere Schritte unterteilt, welche dann von unterschiedlichen Knoten bearbeitet wird. Für die Zerlegung von den Aufgaben gibt es zwei Ansätze:

- Die Berechnung wird in mehrere kleine Schritte zerteilt und diese werden dann an einzelnen Knoten weitergegeben, welche die Berechnung parallel ausführen.
- Ein Knoten führt die gesamte Berechnung sequenziell durch, jedoch immer nur für einen Datensatz.

Die Aufteilung und Zuordnung der Aufgaben erfolgt über ein *Job Management System*. Ein weitere wichtige Voraussetzung für HPC-Cluster ist, dass die Knoten über ein Netzwerk miteinander verbunden sind.

Das Einsatzgebiet von so einem Cluster liegt meist in einem wissenschaftlichen Bereich, wo viele Berechnungen durchgeführt werden müssen.

- **Load Balancing-Cluster (LB-Cluster)**

Load Balancing-Cluster haben ein ähnliches Ziel wie HPC-Cluster, nämlich, die Last auf mehrere Maschinen aufzuteilen. LB-Cluster kommen meistens bei Webservern zum Einsatz, wo sie dafür verwendet werden, einen Flaschenhals bei einem Dienst zu verhindern. Ein häufiges Verfahren von einem LB-Cluster, ist das Vorschalten eines Load Balancers oder Frontend-Server, der die Anfragen aufteilt und den Knoten zuweist. So kann es nicht geschehen, dass ein Gerät einen Flaschenhals verursacht.

Bei Load Balacing-Cluster werden selten die einzelnen Knoten aufgewertet, weil sie meistens keine großen Rechenaufgaben oder ähnliches bearbeiten müssen, sondern es werden einfach neue Knoten hinzugefügt, wenn mehr Leistung benötigt wird. Häufig werden für so einen Cluster auch nur COTS¹ -Komponenten verwendet, das heißt es werden keine teuren Spezialcomputer verbunden, sondern preisgünstige Standardcomputer.

- **Storage-Cluster (SC)**

Storage-Cluster verfolgen dasselbe Ziel wie ein HA-Cluster, jedoch liegt der Unterschied darin, dass bei Storage Cluster die Knoten keine Rechner oder Server sind, sondern Speichermedien, bei denen die Verfügbarkeit möglichst hoch sein muss.

Auch bei den SC gibt es wieder eine Unterteilung in Failover und Takeover (siehe 5.2), sowie "Stretched Cluster".

Dieser Sachverhalt wird in (LR:Web01, vgl. fujitsu.com 31.10.2020) genauer beschrieben.

Mehr Informationen über die unterschiedlichen Arten von Clustern findet man auf diesen beiden Seiten. (LR:Web02, vgl. it-administrator.de 31.10.2020) (LR:Article01, vgl. datacenter-insider.de 31.10.2020)

¹Commercial off-the-shelf

5.3 Grenzen der einfachen Cluster

5.3.1 Amdahl's Law

Wenn man über Optimierung durch Parallelisierung wie bei Clustersystemen spricht, kommt man nicht an 'Admahl's Law' vorbei. Diese Richtlinie für die Optimierung von Programmen, durch gleichzeitiges Ausführen von Programmteilen, entstand aus dem Paper: *Validity of the Single Processor Approach to Achieving Large-Scale Computing Capabilities*. Es wurde erstmals auf der AFIPS spring joint computer conference² von dem damaligen IBM Mitarbeiter Gene M. Amdahl vorgestellt.

For over a decade prophets have voiced the contention that the organization of a single computer has reached its limits and that truly significant advances can be made only by interconnection of a multiplicity of computers in such a manner as to permit cooperative solution. Variously the proper direction has been pointed out as general purpose computers with a generalized interconnection of memories or as specialized computers with geometrically related memory interconnections and controlled by one or more instruction streams.

(LR:Paper01, Validity of the Single Processor Approach to Achieving Large-Scale Computing Capabilities. S.483-485, 30.12.2020)

Dieses Gesetz besagt, dass der Geschwindigkeitsgewinn, wenn man ein Problem durch parallele Verarbeitung löst, durch die Formel

$$\text{Speedup} = \frac{1}{r_s + \frac{r_p}{n}}$$

angegeben wird. Wobei ' r_s ' für den Teil der Lösung steht, der sequenziell abgearbeitet werden muss und ' r_p ' für den Teil der parallel durchgeführt werden kann; zusammen ergeben die beiden Werte 1 und stehen für den gesamten Aufwand der Lösung. n steht für die Anzahl von Einheiten der parallelen Bearbeitung.

²Sie wurde 1967 abgehalten

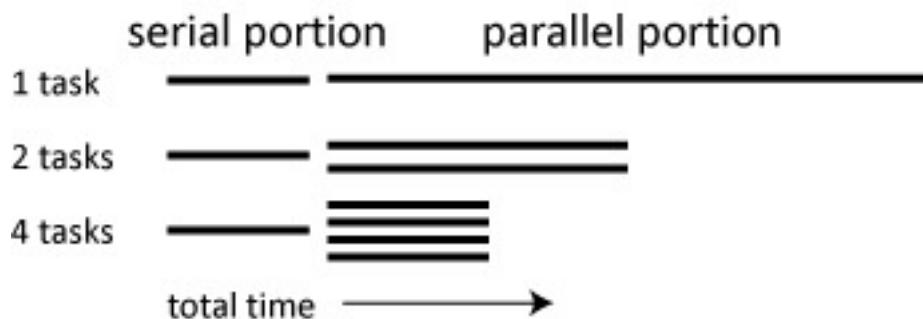


Abbildung 5.1: Optimierung durch Parallelisierung
Quelle: (LR:Web27, vgl. cornell.edu 23.03.2021)

Das heißt, die Zeit, welche man durch Parallelisierung optimieren kann, wird von dem sequenziellen Teil der Lösung limitiert.

Das hat zur Folge, dass es eine Obergrenze an dem Geschwindigkeitsgewinn durch Parallelisierung gibt. Diese Grenze variiert je nachdem, wie groß der sequenzielle Teil ist. Dadurch ergibt sich die folgende Grafik.

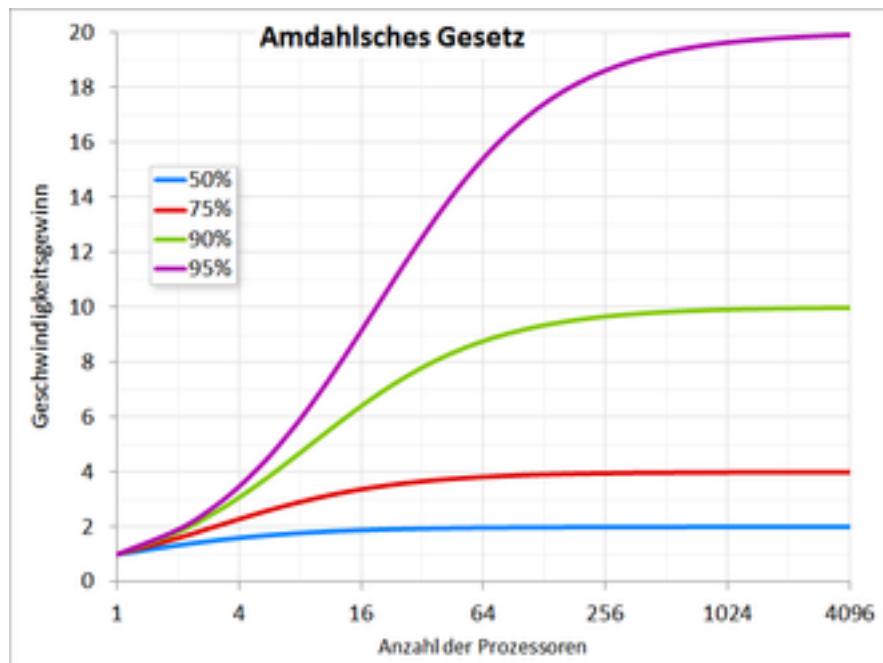


Abbildung 5.2: Geschwindigkeitsgewinn beim Parallelisierung
Quelle: (LR:Web23, vgl. researchgate.net 21.03.2021)

Ausfall in einem Cluster

Ein weiteres Problem bei Clustercomputer tritt auf, wenn es zum Ausfall eines Knoten kommt oder wenn dieser überlastet ist. Dadurch kann es passieren, dass alle anderen Einheiten mit dem Lösen ihres eigenen Teilergebnisses bereits fertig sind, jedoch ein Knoten ausgefallen ist oder für das Lösen mehr Zeit benötigt. Folglich, kann das Endergebnis nicht fertiggestellt werden.

In moderneren Implementationen von Cluster Verarbeitung³ wird dieses Problem damit umgangen, dass eine Teilaufgabe häufig an mehrere Knoten übergeben wird oder, dass durch Monitoring- und Mapping-Algorithmen ein Ausfall oder Überlastung erkannt werden kann.

³wie Apache Spark oder Hadoop

Kapitel 6

Big Data Frameworks, die mit Clustersystem arbeiten

Big Data Frameworks, die mit Clustersystem arbeiten, sind Gerüste, die es ermöglichen, vorgefertigte Umgebungen und Methoden zu verwenden, um ein Clustersystem zu erstellen und zu steuern.

6.1 Apache Hadoop

Apache Hadoop ist eine verteilte Big Data Plattform, die von Google basierend auf dem Map-Reduce Algorithmus entwickelt wurde, um rechenintensive Prozesse bis zu mehreren Petabytes zu erledigen.

(LR:Article02, datasolut.com 1.1.2021)

Apache Hadoop ist ein Open Source Framework, welches in Java entwickelt wird. Es wird von der Apache Software Foundation betreut und ist auch das Top-Level-Projekt dieser Foundation. Es kann dazu verwendet werden, um große Mengen an Daten zu speichern und zu bearbeiten, das funktioniert durch die Aufteilung der Arbeiten auf mehrere Hardwareträgern, durch den Map-Reduce Algorithmus, welcher von Google Inc. entwickelt¹ wurde. Es wird von sehr vielen großen Firmen eingesetzt, zum Beispiel: Facebook, IBM, Adobe und eBay. Windows hat es auch in Windows Azure und SQL Server 2019 integriert.

¹Seit 2010 von Google Inc. patentiert

6.1.1 Geschichte

- **2002** Doug Cutting und Mike Cafarella arbeiteten an Apache Nutch, ein Projekt zum Entwickeln einer Suchmaschine, welche über 1 Milliarde Seiten durch Indexe verwalten sollte. Die Implementation wäre jedoch zu teuer gewesen.
- **2004** Mit Hilfe der beiden von Google veröffentlichten Paper zu GFS²³ und Map-Reduce Algorithmus, hatten Doug Cutting und Mike Cafarella die Antworten auf ihre beiden Probleme: das Speichern großer Datensätze und die Verarbeitung dieser. Somit begannen sie die Implementierung unter einer Open Source Lizenz.
- **2005** Cutting bemerkte, dass Nutch nur auf 20 bis 40 Knoten stabil laufen könnte, um das Problem jedoch zu beheben wären mehrere Leute notwendig, deshalb begannen die beiden nach einer Firma zu suchen, welche sie bei der Entwicklung unterstützen kann.
- **2006** Nutch tritt Yahoo bei und es wird unbenannt in Hadoop (nach dem Namen eines gelben Spielzeugelefanten, von Cuttings Sohn, deshalb ist auch das Logo ein gelber Elefant). Das Ziel von Hadoop war es, ein zuverlässiges und skalierbares Open-Source Framework zu liefern.
- **2007** Yahoo testete Hadoop erfolgreich auf einem 1000-Knoten-Cluster.
- **2008** Im Jänner wurde Hadoop durch die ASF⁴ offiziell veröffentlicht und im Juli des Jahres wurde es auf einem 4000-Knoten-Cluster erfolgreich getestet.
- **2009** Cutting verließ Yahoo, damit sich Hadoop auch in andere Segmente verbreiten kann. Außerdem wurde ein PB⁵ an Daten in weniger als 17 Stunden sortiert und indexiert.
- **2011** Die ASF veröffentlichten Apache Hadoop Version 1.0.
- **2013** Die neue Version 2.0.6 wurde veröffentlicht.
- **2017** Die Apache Software Foundation veröffentlichte die aktuellste⁶ Version Apache Hadoop 3.0.

(LR:Article05, vgl. [geeksforgeeks.org 9.1.2021](https://geeksforgeeks.org/9.1.2021))

²³Google File System

³Erschienen 2003

⁴Apache Software Foundation

⁵PetaByte

⁶Jänner 2021

6.1.2 Funktionsweise

Die Funktionen von Hadoop teilen sich in drei Hauptteile auf HDFS⁷, YARN⁸ und dem Map-Reduce Algorithmus. Jedoch sind auch die zahlreichen Erweiterungen⁹ von Hadoop sehr wichtig für viele Funktionalitäten und können daher auch als eigener Bestandteil angesehen werden.

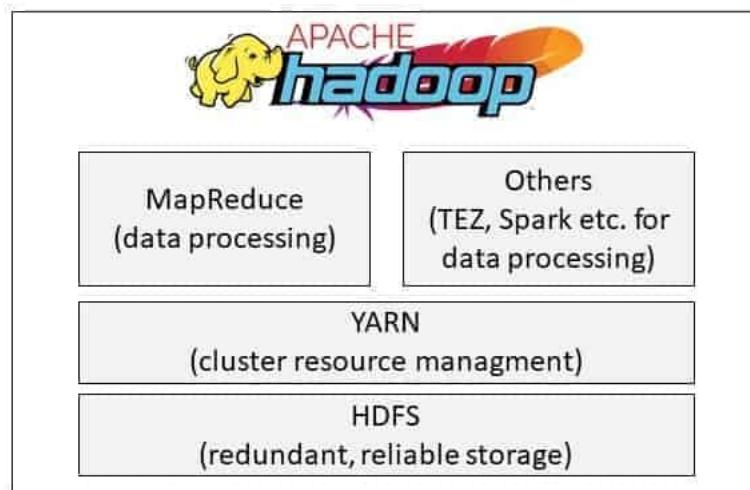


Abbildung 6.1: Komponenten von Hadoop
Quelle: (LR:Article02, vgl. datasolut.com 02.01.2021)

HDFS

The Hadoop Distributed File System (HDFS) is a distributed file system designed to run on commodity hardware. It has many similarities with existing distributed file systems. However, the differences from other distributed file systems are significant. HDFS is highly fault-tolerant and is designed to be deployed on low-cost hardware. HDFS provides high throughput access to application data and is suitable for applications that have large data sets. HDFS relaxes a few POSIX requirements to enable streaming access to file system data. HDFS was originally built as infrastructure for the Apache Nutch web search engine project. HDFS is now an Apache Hadoop subproject.
(LR:Docs01, hadoop.apache.org 2.1.2021)

⁷Hadoop Distributed File System

⁸Yet Another Resource Negotiator

⁹In der Grafik als 'Other'

Architektur

HDFS arbeitet mit einer Master/Slave Architektur das heißtt, es gibt einen 'Namenode', der als Master-Knoten fungiert. Dieser ist für die Verwaltung des Dateisystems und der Zugriffe aller Benutzer verantwortlich. Dann gibt es die 'Datanodes', diese sind die Slave-Knoten und dienen als Speicher.

Intern werden die Daten in 'Blocks' aufgeteilt. Dabei übernimmt der 'Namenode' die Aufgaben für das Öffnen, Schließen und Umbenennen von Dateien und Ordnern und die 'Datanodes' übernehmen die Lese- und Schreibzugriffe, sowie das Erstellen, Löschen und Replizieren, wenn es von der 'Namenode' verlangt wird.

Für den Client scheint HDFS als normales Dateisystem auf, indem er Dateien und Ordner verwalten kann, die Aufteilung in Datenblöcken geschieht intern.

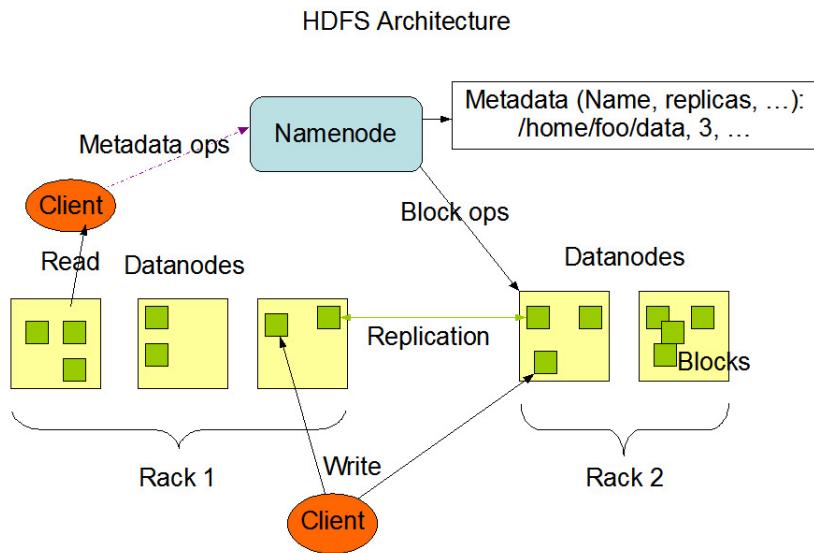


Abbildung 6.2: Die Architektur von HDFS
Quelle: (LR:Article15, vgl. opensource.com 21.03.2021)

Die 'Namenode' oder 'Datanode' sind nur Teile einer Java Software, die auf jedem Rechner mit Linux oder GNU als Betriebssystem laufen, somit kann dieses Dateisystem auch auf einfachen Rechnern oder Servern genutzt werden. Normalerweise gibt es in einem Cluster immer einen 'Namenode' und mehrere Systeme wo eine Instanz von einem 'Datanode' läuft.

Der Einsatz von einem einzigen Master-Knoten vereinfacht die Architektur, es ist jedoch auch möglich mehrere Systeme als 'Namenode' einzustellen. Das System ist aber so entwickelt worden, dass keine Daten durch den 'Namenode' gehen, daher ist meistens ein System ausreichend.

Redundanz

Ein wichtiger Teil von HDFS ist seine Zuverlässigkeit über einen großen Cluster. Die Files werden als eine Menge von Blöcken (alle bis auf den letzten sind gleich groß) abgespeichert. Diese Blöcke werden zur Abwehr von Fehler redundant auf mehreren 'Datanodes' gespeichert.

Die Anzahl, wie oft ein File abgespeichert werden soll, wird, 'replication factor' genannt und kann für jede Datei konfiguriert werden. Man sollte jedoch bedenken, dass umso höher dieser Faktor ist, desto höher ist auch der Lese- und Schreibzugriff.

Die 'Namenode' bestimmt, wo diese Blöcke abgespeichert werden. Dazu bekommt sie regelmäßig einen 'Heartbeat' von jeder funktionierenden 'Datanode', der aussagt, dass sie noch funktioniert. Schickt eine 'Datanode' keinen Herzschlag, wird sie von der 'Namenode' als tot angesehen.

Stärken	Schwächen
verwalten von Datenmengen bis zu PBs (PetaBytes)	intransparente Verteilung von Blöcken
skalierbar auf tausenden Server	manche Dienste sind durch das HDFS angezeigt
automatische Redundanz der Daten	
sehr hohe Fehlertoleranz	

Tabelle 6.1: Vor- und Nachteile von HDFS

Viele Projekte verwenden nicht mehr HDFS als Dateisystem, sondern steigen um auf intelligenten Cloudspeicher wie (AWS S3 oder Azure BlobStore), da diese meist billiger und einfacher zu bedienen sind.

YARN

The fundamental idea of YARN is to split up the functionalities of resource management and job scheduling/monitoring into separate daemons¹⁰. The idea is to have a global ResourceManager (RM) and per-application ApplicationMaster (AM). An application is either a single job or a DAG¹¹ of jobs. (LR:Docs02, hadoop.apache.org 3.1.2021)

Neben den beiden schon erwähnten Resourcemanager und Applicationmaster gibt es noch einen Nodemanager. Der Nodemanager ist ein Framework, welches auf jeden

¹⁰Computerprozess

¹¹Ein gerichteter Graph ohne eine Schleife darin

einzelnen Knoten selbst läuft. Er ist zuständig für das Überwachen der Container innerhalb eines Knoten, das heißt, er überwacht die verwendeten Ressourcen in einem Node. Diese Daten übergibt er dann dem Resourcemanager.

Der Resourcemanager ist die oberste Instanz, in dem Framework und verwaltet die Ressourcen zwischen allen Anwendungen und Systemen in dem Cluster.

Jede Anwendung hat einen eigenen Applicationmaster, dieser ist eigentlich nur eine Library in dem Framework und ist damit beauftragt, gemeinsam mit dem Resourcemanager die Ressourcen zu vereinbaren, dazu arbeitet es mit dem Nodemanager zusammen. Weiteres arbeitet der Applicationmaster auch deswegen mit dem Nodemanager zusammen, um seine Jobs auszuführen und zu überprüfen.

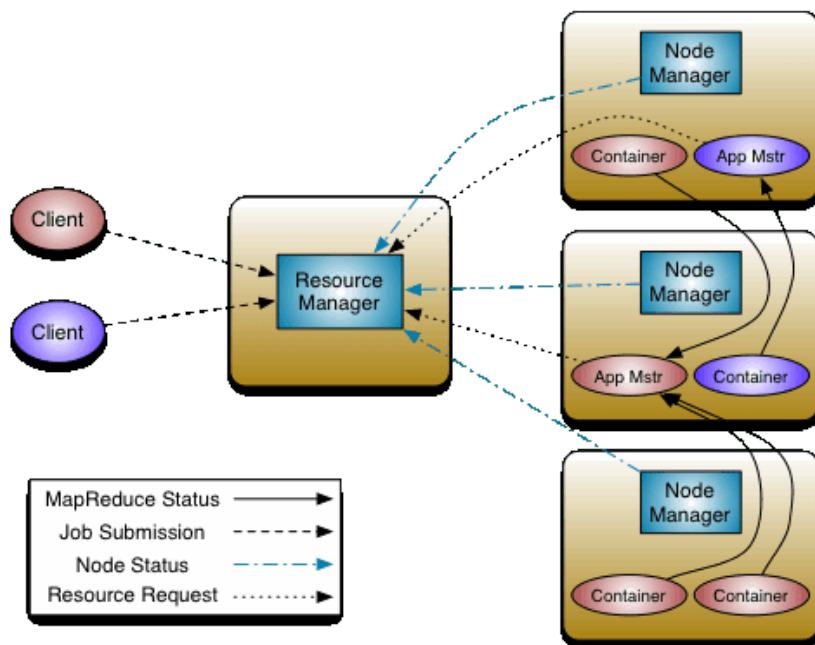


Abbildung 6.3: Die Architektur von YARN
Quelle: (LR:Web28, vgl. ionos.co.uk 23.03.2021)

MapReduce

Der MapReduce-Algorithmus, ist der Hauptteil von Hadoop und auch der Grund dafür, dass Hadoop sich auf einen Cluster, mit mehreren tausend Knoten, skalieren kann. Der Algorithmus lässt sich, wie der Name schon sagt, auf zwei Schritte reduzieren: Mapping & Reduzieren.

Der erste Schritt, Mapping, findet auf den Datanodes statt, indem jeder Datanode seine eigenen Daten in einer vorgegebenen Art und Weise selektiert, meistens entstehen daraus Tupeln¹².

¹²Schlüssel/Wert Paare

Die Tupeln werden dann im Rahmen des zweiten Schrittes, Reduce, zum Namenode geschickt, dieser fasst alle Daten, die er bekommt, dann zusammen. Dadurch ist es möglich, über den Namenode Analysen zu betreiben.

Somit ist MapReduce in der Lage, die unstrukturierten Daten auf den einzelnen Knoten des Cluster in ein Format zu bringen, damit diese dann ausgewertet werden können. Das funktioniert dadurch, dass MapReduce kein fest vorgeschriebener Algorithmus ist, sondern nur eine Art von Framework. In welchem über Programmiersprachen, meist Java, Scala oder Python, die genauen Schritte definiert werden können.

Ein gutes Beispiel von so einem selbst geschriebenen Programm finden Sie hier: (LR:Article03, data-science-blog.com 3.1.2021)

Hier wurde über Java ein Word-Count-Algorithmus für die MapReduce-Logik in Hadoop geschrieben.

Combine

Häufig reicht das Mapping auf den einzelnen Datanodes nicht aus, um den Namenode so viel Last abzunehmen, dass es zu einem effizienten Weg der Datenaufbereitung kommt. So wird zusätzlich zu dem parallel laufenden Map Prozess noch ein Combine Prozess auf den Datanodes ausgeführt. Das Ziel ist es, dem Namenode möglichst viel Arbeit abzunehmen.

Der Combine Schritt ist dazu gedacht, dass der Datanode lokal ein Reduce auf seinen eigenen Daten durchführt, dann muss der Namenode nur aus allen Ergebnissen ein Endergebnis formen.

Beispiel

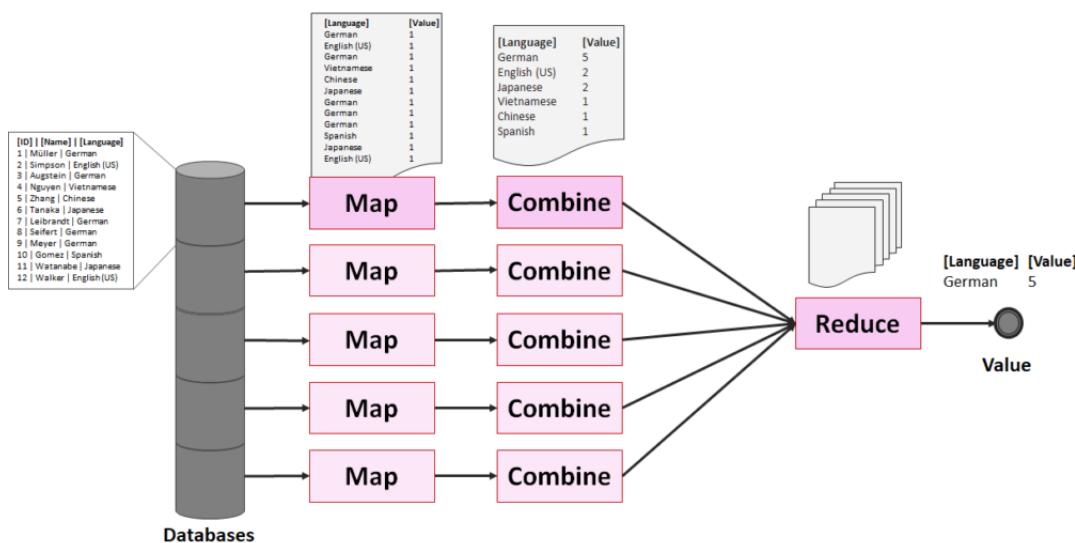


Abbildung 6.4: Ablauf von MapReduce
Quelle:(LR:Article03, vgl. data-science-blog.com 3.1.2021)

Auf der Grafik ist ein Word-Count durch ein MapReduce durchgeführt worden. Mit Hilfe von Select-Statements wird dargestellt, welche Daten in einem Schritt wie behandelt werden. Man kann auch erkennen, was für eine wichtige Rolle das Hinzufügen von

einem Combine Prozesse spielt, wenn man bedenkt, dass ansonsten der Namenode diesen Schritt für alle Knoten übernehmen müsste.

Kritik

Auch wenn der MapReduce-Algorithmus, viele Vorteile, bezogen auf einen großen Cluster hat, gilt er bereits als veraltet, dadurch wird er immer häufiger durch DAG¹³(siehe 6.1.3) basierte Algorithmen für den Zugriff ersetzt.

Ein weiteres Problem vom MapReduce-Algorithmus liegt darin, dass umfangreiche Analyseabfragen nur sehr umständlich behandelt werden können, denn für ein einfaches Join in der Abfrage benötigt man bereits mehrere Ketten aus MapReduce-Algorithmen. Das hat zur Folge, dass sich MapReduce nicht dazu eignet, um maschinelles Lernen mit den Daten durchzuführen.

(LR:Article03, vgl. data-science-blog.com 4.1.2021)

6.1.3 Erweiterungen

Hadoop Ecosystem

Ein Apache Hadoop Ecosystem ist die Menge von Add-On's¹⁴, die man zusätzlich zu den einfachen Apache Hadoop Bausteinen, welche im Kapitel '2.1.2 Funktionsweise' bereits beschrieben wurden, auf sein System ladet. Diese Möglichkeit der Erweiterung des Hauptprogrammes ist einer der wichtigsten Punkte, warum Apache Hadoop zu einem der besten Implementierungen im Bereich Clustersystemen gehört.

Viele dieser Erweiterungen werden auch häufig schon als Standard-Komponente von Hadoop angesehen, weil sie entweder bekannte Schwachstellen von Hadoop beseitigen, den Umgang für den Benutzer als auch für das Verwalten von Hadoop um einiges vereinfachen oder die Performance verbessern.

Die unten angegebene Grafik ist nicht die absolute Lösung für den Einsatz von diesen Erweiterungen in Apache Spark, sie ist jedoch eine sehr weit verbreitete Lösung und auch eine der Besten, welche viele bekannte Erweiterungen beinhaltet.

Die hellblau eingefärbten Boxen sind jene, die bereits im vorigen Kapitel beschrieben wurden und Teil von Apache Hadoop sind.

Die restlichen Blöcke sind optionalen Erweiterungen, mit denen man die Funktionen erweitern und die Performance von Apache Hadoop verbessern kann.

¹³Directed-Acyclic-Graph

¹⁴Erweiterungen

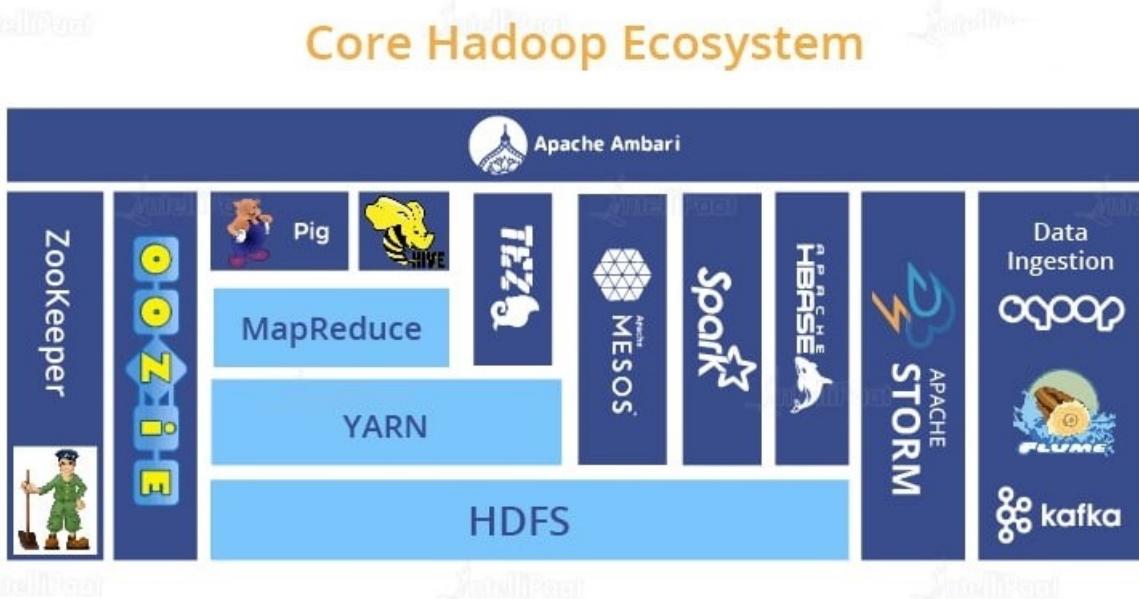


Abbildung 6.5: Apache Hadoop Ecosystem
Quelle: (LR:Web11, vgl. intellipaat.com 9.1.2020)

Zookeeper

ZooKeeper is a coordination service for distributed applications with the motto SZooKeeper: Because Coordinating Distributed Systems is a Zoo. The ZooKeeper framework was originally built at Yahoo. It runs on JVM (Java virtual machine). A few of the distributed applications that use Zookeeper are Apache Hadoop, Apache Kafka, and Apache Storm.

(LR:Web07, edureka.co 4.1.2021)

ZooKeeper funktioniert am besten, wenn es in einem Cluster arbeitet, in dem die 'Read'-Anfragen die 'Write'-Anfragen übertreffen, am besten in einer 10:1¹⁵ Rate. Damit ZooKeeper auf einem Cluster läuft, muss er über alle Knoten den Status abfragen können. ZooKeeper ist auch sehr fehlerresistent das heißt, wenn es zu einem Problem im Cluster kommt, ist er einer der letzten Prozesse die ausfallen.

¹⁵10 'Reads' zu einem 'Write'

Funktionen von ZooKeeper:

- **Naming service** Die Knoten innerhalb eines Systems können mit Namen ange- sprochen werden. Das funktioniert ähnlich wie DNS¹⁶.
- **Configuration management** Wenn ein neuer Knoten im Cluster hinzugefügt wird, wird das System auf den neusten Stand konfiguriert.
- **Cluster management** Es liefert eine Übersicht von allen Knoten und deren Status in einem Cluster.
- **Leader election** Es kann ein Knoten als Hauptknoten festgelegt werden.

Hive

The Apache Hive data warehouse software facilitates reading, writing, and managing large datasets residing in distributed storage and queried using SQL syntax.

(LR:Web08, cwiki.apache.org 6.1.2021)

Apache Hive erstellt ein DWH¹⁷ für Apache Hadoop, dieses Data Warehouse kann dann SQL Abfragen verarbeiten, somit ist es möglich, über SQL-Befehle auf die Daten im HDFS zuzugreifen.

Mit HiveQL, die Sprache für Hive, die ähnlich SQL_1999 ist, können Anwender oder Programme, Anfragen an das Data Warehouse senden, diese SQL Anfragen werden dann in MapReduce-Jobs oder einen DAG (siehe 6.1.3) umgewandelt.

Eine ausführliche Dokumentation von HiveQL findet man hier: (LR:Web09, vgl. cwiki.apache.org 6.1.2021).

Hive ist drauf ausgelegt, zu skalieren (damit es auf den ganzen Hadoop Cluster skalieren kann), daher ist es auch sehr performant, fehlertolerant und an kein festes Format der Daten gebunden. Es ist jedoch nicht drauf ausgelegt OLTP¹⁸ zu verarbeiten, es ist nur möglich die traditionellen DWH Aufgaben durchzuführen, wie das Lesen, Auswerten oder Verwalten von großen Datensets.

Eine gute Anleitung wie man Apache Hive in seinem Apache Hadoop installiert findet man hier: (LR:Web10, cwiki.apache.org 6.1.2021)

¹⁶Domain Name System

¹⁷Data Warehouse

¹⁸Online Transaction Processing

HBase

Apache HBase, ist eine verteilte Datenbank, welche darauf ausgelegt ist, strukturierte Daten in Tabellen, mit Millionen von Spalten mit Millionen von Zeilen, zu verwalten. HBase ist dabei eine skalierbare, verteilte NoSQL Datenbank, welche auf das HDFS aufgesetzt wird. Es bietet die Möglichkeit für real-time¹⁹ Lese- und Schreibzugriffe auf die Daten im HDFS.

Eine Liste mit allen Fähigkeiten von HBase findet man auf der Seite von Apache HBase: (LR:Docs03, hbase.apache.org 8.1.2021)

Intern gibt es bei HBase zwei unterschiedliche Komponenten:

- **HMMaster** ist kein Teil des eigentlichen Datenspeichers, er verwaltet die Auslastungen aller Region Server. Dabei hat er folgende Aufgaben: Überwachen und Verwalten des Cluster, Administration von HBase (Erstellen, Update oder Löschen von Tabellen), Kontrolle der Ausfallsicherung und Verarbeitung der DDL Anfragen
- **Region Server** Er ist der Slave-Knoten im System, welcher für die Durchführung von Lese-, Schreibe-, Update- und Delete-Anfragen zuständig ist. Er läuft auf jeden DataNode im Hadoop Cluster.

Im (LR:Article04, data-flair.training 6.1.2021) von 'data-flair.training' findet man eine ausführliche Verteilung der Aufgaben, außerdem eine Anleitung zur Einbindung in Hadoop und zum Arbeiten mit HBase.

Tez

Tez: A generalized data-flow programming framework, built on Hadoop YARN, which provides a powerful and flexible engine to execute an arbitrary DAG of tasks to process data for both batch and interactive use-cases. Tez is being adopted by Hive, Pig and other frameworks in the Hadoop ecosystem, and also by other commercial software (e.g. ETL tools), to replace Hadoop MapReduce as the underlying execution engine.

(LR:Docs04, hadoop.apache.org 6.1.2021)

Tez ist also eine verbesserte Version vom MapReduce-Algorithmus, und ersetzt diesen dadurch. Der größte Unterschied liegt darin, dass Tez mit DAG arbeitet.

Die zwei Hauptziele von Tez sind:

¹⁹Abfragen wie es genau jetzt ausschaut

- Dem Benutzer mehr Anwendungsmöglichkeiten bieten:
 - Es gibt API's, womit Tez gesteuert werden kann.
 - Flexibles Input-Processor-Output Laufzeitmodell, das heißtt, es wird dynamisch nach dem besten Input, Prozessoren und Output gesucht und es geschieht nicht alles auf einem Knoten.
 - Data type agnostic heißtt, dass es egal ist wie die zugrundeliegenden Daten ausschauen, Tez kann sie trotzdem verarbeiten.
 - Das Einsetzen von Tez in Hadoop ist vereinfacht.
- Verbesserung der Performance von Abfragen.
 - Es arbeitet Performanter als MapReduce.
 - Die zur Verfügung stehenden Ressourcen werden optimal genutzt.
 - Es wird bereits während es läuft schon die Rekonfiguration geplant.
 - Der physische Datenfluss über die Knoten wird dynamisch, nach der Auslastung, gewählt.

(LR:Docs04, vgl. hadoop.apache.org 7.1.2021)

DAG

DAG steht für, 'Directed acyclic graph' (de. gerichteter azyklischer Graph), in einem solchen Graph gibt es Ecken und gerichtete Verbindungen zwischen den Ecken, diese nennt man Kanten. Tez baut so aus jeder Anfrage einen Graph zusammen, wobei eine Ecke dabei ein Teil von dieser Abfrage oder Script ist. Die Kanten geben an, welche Funktionen als Vorgänger bzw. Nachfolger gelten. Das hat den Vorteil, dass es möglich ist, Teile parallel abzuarbeiten, bei MapReduce können einzelne Teile der Abfrage nur sequenziell ablaufen.

Weitere Erweiterungen

- **Apache Ambari** ist ein webbasiertes Tool, welches zur Verwaltung und Überwachung von Hadoop Cluster verwendet werden kann. Dabei stellt es Funktionen wie eine 'Cluster Health' oder heatmaps zur Verfügung.
- **Oozie** ist eine Art, wie man seine Tasks innerhalb eines Clusters planen kann.
- **Apache Pig** ist eine datenstromorientierte Programmiersprache, kann dafür verwendet werden, um einen kontinuierlichen Datenstrom zu verwalten.

- **Mesos** kann die Aufgaben von YARN übernehmen, der größte Unterschied ist, dass der Mesos Master alle Aufgaben bekommt und dann an den Datanode weiter gibt.
- **Apache Spark** kann auch für die Abfragen verwendet werden (siehe 6.2.2).
- **Apache Strom** kann dazu verwendet werden um Streams zu verarbeiten.
- **Sqoop** wird dazu verwendet um Daten zwischen Hadoop und einer relationalen Datenbank zu übermitteln.
- **Flume** wird zur Verwaltung von Logdaten genutzt, dabei bietet es Möglichkeiten große Menge von Logdaten zu verschieben, zu aggregieren und zu sammeln.
- **Apache Kafka** funktioniert ähnlich wie Flume, nur, dass es über Echtzeitdaten Pipelines arbeitet.

(LR:Web11, vgl. intellipaat.com 9.1.2020)

6.1.4 Hadoop-Cluster

Ein Hadoop-Cluster beschreibt das Zusammenarbeiten von mehreren Servern oder Computern, die durch eine Apache Hadoop Instanz verwaltet werden. Innerhalb dieses Clusters erfolgt die Aufteilung in die bereits bekannten Rollen: Namenode und Datanode, jedoch gibt es auch Clientnodes. Diese Clientsnodes sind für die Zugriffe der Benutzer zuständig.

6.1.5 Fazit

Auch wenn Apache Hadoop mittlerweile schon sehr alt ist, war es trotzdem das erste Projekt, welches ein stabiles Framework liefert hat, um mit Clustersystem zu arbeiten. Das ist auch der Grund, weshalb es auch jetzt noch immer eine sehr weit verbreitete Lösung ist.

Man muss aber auch bedenken, dass Hadoop eindeutige Schwachstellen hat, die andere Big Data Frameworks nicht haben. Etwas was man dabei nicht weglassen kann ist die Tatsache, dass es mit Apache Hadoop nur sehr bedingt möglich ist, maschinelles Lernen durchzuführen. Aber auch der Umgang mit sehr stark verknüpften Daten kann zu einem Problem führen, da die Abfragen schnell unübersichtlich und kompliziert werden.

In diesen Bereichen gibt es Alternativen, die eindeutig besser sind als Hadoop.

Sollte man aber ein Projekt entwickeln, welches diese Aufgabenstellungen nicht bearbeitet, ist Hadoop noch immer eine der besten Lösungen, weil durch die große Menge von Erweiterungen, der Umgang damit vereinfacht werden kann und Funktionen an sein eigenes Projekt angepasst werden können. Dazu gibt es auch eine sehr ausführliche Dokumentation von Hadoop und seinen Erweiterungen und viele Beispielprojekte. Daher kann man abschließend sagen: Apache Hadoop kann nicht mit anderen Frameworks mithalten, wenn es um moderne Aufgabenstellungen wie AI geht, werden diese Aufgabenstellungen jedoch nicht benötigt, ist es noch immer das stabilste und vor allem am besten skalierbare Framework.

6.2 Apache Spark

Spark is an open-source parallel processing framework for running large-scale data analytics applications across clustered computers, i.e., it provides limited in-memory data storage that supports the reuse of data on distributed collections in an application array. It does not include a data management system and is therefore usually deployed on top of Hadoop or some other storage platform.

(LR:Article06, gigaspaces.com 03.02.2021)

Apache Spark ist eine unified analytics engine (de. Einheitliche Analyse-Engine), welche auf die Verarbeitung von großen Datenmengen ausgelegt ist. Es wird so wie Apache Hadoop von der Apache Software Foundation betreut.

Dabei kann es bis zu 100-mal schneller die Daten als Hadoop verarbeiten, indem es den modernsten DAG²⁰ Algorithmen, einen Anfragenoptimierer und eine Execution Engine verwendet. Mehr dazu jedoch unter 2.2.3 Funktionsweise von Apache Spark. Eine Apache Spark Applikation lässt sich einfach über Python, Java, Scala, R oder sogar SQL steuern.

Außerdem verbindet es noch viele Aufgabenbereiche miteinander. Zum Beispiel das Abspeichern von Daten über DataFrames oder SQL Abfragen, sowie Machine Learning und die grafische Aufbereitung sind möglich.

(LR:Docs06, vgl. spark.apache.org 05.02.2021)

Verbreitung

Man kann Apache Spark entweder als einzelnes Programm verwendet oder es in Verbindung mit einer der vielen möglichen Frameworks oder Datenspeicher²¹ anwenden. Innerhalb einer solchen Vereinigung kann Apache Spark dazu verwendet werden, um den Datenzugriff zu beschleunigen. Aufgrund dessen wird Apache Spark auch in viele großen Firmen wie Yahoo, eBay oder Netflix verwendet.



Abbildung 6.6: Apache Spark Verbreitung
Quelle: (LR:Web12, vgl. databricks.com 01.02.2021)

²⁰siehe 6.1.3

²¹siehe 6.6

6.2.1 Geschichte

Matei Zaharia startete **2009** die Entwicklung von Spark in der UC Berkeley, anschließend wurde das Programm **2010** unter einer Open Source Lizenz veröffentlicht.

2013 wurde es an die Apache Software Foundation übergeben und Matei Zaharia und sein Entwicklerteam haben Databricks gegründet. Das Unternehmen arbeitet noch immer mit an der Entwicklung von Apache Spark.

2014 hat Databricks mit Apache Spark einen Rekord für das Sortieren von großen Datenmengen aufgestellt, indem sie 100 TB²² in 24 Minuten verarbeitet haben. In demselben Jahr wurde Spark auch zu einem Top-Level Apache Projekt.

Derzeit²³ haben über 1000 Entwickler von mehr als 300 unterschiedlichen Organisationen an der Entwicklung teilgenommen. Die Funktionen von Apache Spark werden in Research-Papers vorgestellt. In der Dokumentation findet man eine Liste dazu: (LR:Paper02, spark.apache.org 04.02.2021).

6.2.2 Verhältnis zu Hadoop

... Neither can replace the other and in actual fact, Hadoop and Spark complement each other. Both have features that the other does not possess. Hadoop brings huge datasets under control by commodity systems. Spark provides near real-time, in-memory processing for datasets.

(LR:Article06, gigaspaces.com 03.02.2021)

Um das Verhältnis der beiden Apache Projekten zu verstehen, sollte man zuerst genau definieren, wo ihre Unterschiede liegen:

- **Hadoop**

Hadoop ist darauf ausgelegt in Projekten oder Firmen eingesetzt zu werden in welchen eine Echtzeit-Datenbearbeitung nicht verlangt wird. Es ist nämlich dafür entwickelt worden, das durch die Batch Verarbeitung²⁴, große Mengen von unformatierten Daten, wirtschaftlich gut verarbeitet werden können. Das ist nur durch die parallele Abarbeitung auf den einzelnen Nodes möglich.

Außerdem bietet es mit dem HDFS²⁵ eine Möglichkeit, um diese unformatierten über die Datanodes dezentral abzuspeichern.

²²Terabytes

²³Februar 2021

²⁴Ausführung von großen, sich wiederholenden Abfragen

²⁵Hadoop Filesystem

- **Spark**

Apache Spark hingegen ist genau auf diese Echtzeit Abfragen ausgelegt. Spark ermöglicht 'in-memory' Verarbeitung von Daten und das Erzeugen von Daten-Streams, was für Echtzeit Analysen und Abfragen von Daten genutzt werden kann. Spark ist bis zu 100-mal schneller als Hadoop, wenn die Daten im Arbeitsspeicher liegen und 10-mal schneller, wenn sie auf der Festplatte abgespeichert sind.

Außerdem ist durch die unkomplizierte Datenabfrage auch Machine Learning möglich, weswegen es eigene Frameworks (wie MLib) dafür liefert.

Spark liefert jedoch kein eigenes Filesystem, mit dem große Mengen von Daten abgespeichert werden können, wie Hadoop. Es gibt zwar ein eigenes Dataframe System, jedoch um es auf einen Cluster zu verwenden benötigt man dafür ein zusätzliches Framework.

Hadoop und Spark ergänzen sich

Auch wenn Apache Spark in vielen Hinsichten einen Vorteil gegenüber Hadoop bringt wäre es trotzdem falsch zu glauben, dass Hadoop durch Spark abgelöst werden könnte, denn wie es oben aus der Gegenüberstellung schon herauslesbar ist, verfolgen die beiden eigentlich gar nicht dasselbe Ziel. Daher kann man die beiden Apache Top-Level Projekte als Ergänzung zueinander sehen.

Wenn es darum geht, wie man Apache Spark in sein Hadoop System einbinden will, gibt es drei Möglichkeiten:

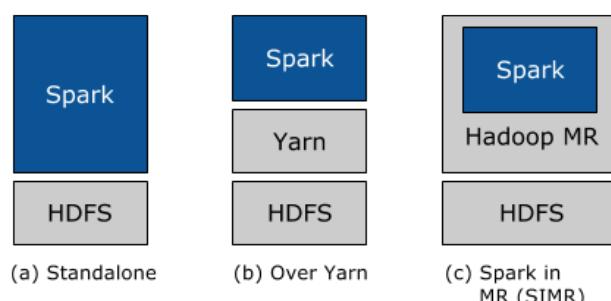


Abbildung 6.7: Einbinden von Apache Spark in Hadoop.
Quelle: (LR:Article07, vgl. databricks.com 04.02.2021)

(a) **Standalone**

Bei dieser Variante läuft neben dem Hadoop System noch eine eigenständige Apache Spark Instanz, mit dieser Instanz ist es dann möglich, Daten aus dem HDFS auszulesen. Dafür können entweder alle Nodes oder es kann nur ein bestimmtes Segment des Clusters durchsucht werden. Es ist die einfachste Variante, weil in Hadoop keine Änderungen durchgeführt werden müssen.

Es ist bei diesem Deployment auch möglich statt Hadoop einen anderen Datenspeicher zu verwenden.

(b) **Over Yarn**

Eine weitere Variante ist es Apache Spark über Hadoop YARN laufen zulassen. Um das durchzuführen, benötigt man weder bestimmte Pakete in Yarn noch Adminrechte innerhalb Hadoop. Diese Methode bietet einen einfachen Weg, Spark auf seinen Hadoop-Cluster einzubinden.

(c) **SIMR²⁶**

Diese Variante bietet die Möglichkeit, Spark Abfragen innerhalb von MapReduce durchzuführen. Die Konfigurationen können binnen Minuten abgeschlossen werden und man kann Anfangen mit Spark Testbefehle ausführen, bevor man eine vollständige Einbindung in den Cluster erledigt.

(LR:Article07, vgl. databricks.com 04.02.2021)

6.2.3 Funktionsweise

Wie in der Einleitung schon angekündigt, handelt es sich bei Apache Spark nicht wie bei Hadoop um ein gesamtes Ecosystem mit dem alles in Bezug auf Cluster Computing und Big Data Verarbeitung bewältigt werden kann, sondern um eine analytics Engine. Diese Engine kann dazu benutzt werden um das Verarbeiten und Aufbereiten von großen Datenmengen zu vereinfachen.

Die Funktionalität setzt sich zusammen aus dem Spark Core und den 4 Hauptkomponenten:

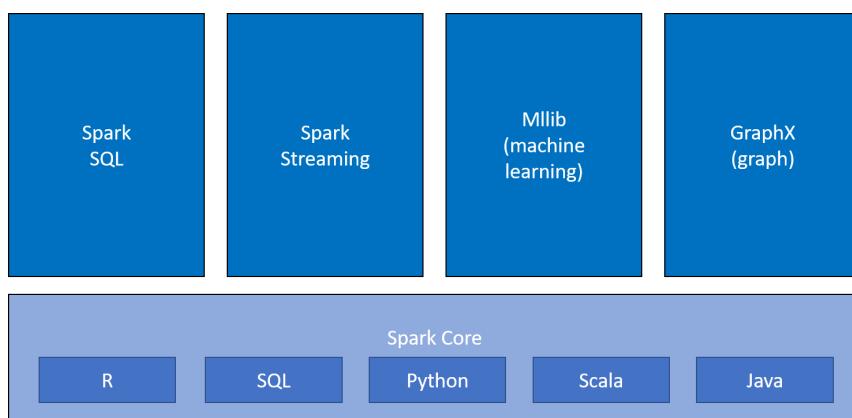


Abbildung 6.8: Hauptbestandteile von Spark
Quelle: (LR:Article08, vgl. datasolut.com 04.02.2021)

²⁶Spark In MapReduce

Spark Core

Der Spark Core bietet die Basis des gesamten Systems, es liefert die Basisfunktion von Spark und ist auch für die Aufgabenaufteilung und Planung zuständig. Der Core steuert auch die I/O²⁷ Funktionalität von Spark.

Neben Basisfunktionalitäten ist der Spark Core auch für die Bereitstellung von den API's²⁸ zuständig. Durch diese API's ist es möglich, Apache Spark durch die Programmiersprachen: R, SQL, Python, Scala und Java zu steuern. Das erhöht die Benutzerfreundlichkeit, da man nicht auf eine Sprache limitiert ist.

RDDs

Als zugrundeliegender Speicher verwendet der Apache Spark Core RDDs²⁹. RDDs setzt sich zusammen aus:

- **Resilient:** Robust; RDDs sind sehr fehlertolerant und sollte es zu einem Hardwareausfall von einem Teil kommen, kann dieser durch die anderen wiederhergestellt werden.
- **Distributed:** Verteilt; die Daten können über mehrere Nodes aufgeteilt werden.
- **Dataset:** Die Daten sind als Dataset abgespeichert.

(LR:Article08, vgl. datasolut.com 04.02.2021)

Kurz gesagt sind RDDs ein Dataset, welches man auf einen Cluster spielen kann.

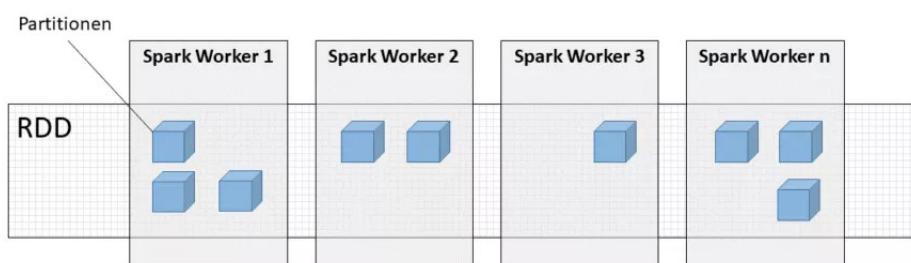


Abbildung 6.9: Spark RDD
Quelle: (LR:Article08, vgl. datasolut.com 05.02.2021)

²⁷In/Out, Ein- und Ausgabe

²⁸Application Programming Interface

²⁹Resilient Distributed Datasets

Spark SQL

Mit Spark SQL können die RDDs aber auch andere Datenspeicher in DataFrames dargestellt werden. Auf diese Datenframes kann man dann über SQL Syntax zugreifen. Somit bietet Spark SQL die Möglichkeit, seine unstrukturierten Daten zu 'semi'-strukturierten Daten umzuwandeln. Gleichzeitig stellt es auch eine Engine zur Verfügung, mit der SQL Befehle auf die Daten abgearbeitet werden können.

Weiteres kann man über Spark SQL auch Hadoop Hive Abfragen ausführen, diese können dann durch die in-memory Speicherung bis zu 100-mal schneller abgearbeitet werden.

Performance

Spark SQL includes a cost-based optimizer, columnar storage and code generation to make queries fast. At the same time, it scales to thousands of nodes and multi hour queries using the Spark engine, which provides full mid-query fault tolerance. Don't worry about using a different engine for historical data.

(LR:Docs07, spark.apache.org 07.02.2021)

Spark Streaming

Apache Spark Streaming ermöglicht durch eine High-Level-API das Verarbeiten von Daten in Echtzeit, dabei werden die Sprachen: Python, Java und Scala unterstützt.

Der Hauptbestandteil ist ein DStream³⁰, der einen normalen Stream in mehrere kleinere Batches aufteilt. Grundlage für diesen DStream bieten die RDDs, weswegen es nicht nur möglich ist diese Streams in Zusammenarbeit mit anderen Spark Libraries zu verwenden, wie MLlib oder Spark SQL, sondern können auch mehrere externe Datenspeicher als Quelle eingesetzt werden. Als Datenquelle können jedoch nicht nur Datenspeicher verwendet werden, welche bereits streaming Daten liefern, sondern es können auch statische Datenquellen herangezogen werden. Beispiele für solche Datenquellen sind: Amazon Kinesis, Kafka, MongoDB und natürlich auch Hadoop. Daher ist es der API auch möglich, Streamingabfragen und Batchabfragen zu verarbeiten.

(LR:Docs08, vgl. spark.apache.org 10.03.2021)

³⁰Discretized Stream

Vorteile von Spark Streaming

Spark gemeinsam mit der Spark Streaming Library, liefert ein System was durch die Spark Engine, sowohl Batchabfragen als auch Streamingabfragen bearbeiten kann. Somit hat man ein Programm, mit dem man alle Anforderungen abdecken kann.

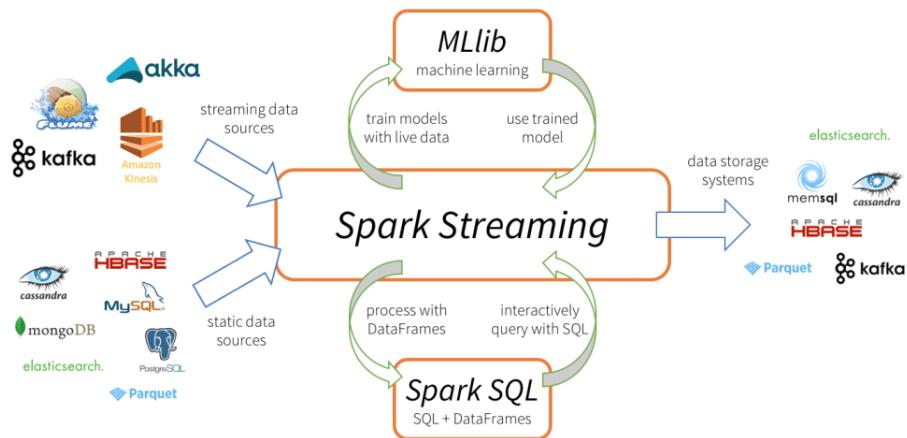


Abbildung 6.10: Veranschaulichung von Spark Streaming
Quelle: (LR:Web13, vgl. databricks.com 03.02.2021)

MLlib (machine learning)

Apache Spark's Machine Learning Library (MLlib) is designed for simplicity, scalability, and easy integration with other tools. With the scalability, language compatibility, and speed of Spark, data scientists can focus on their data problems and models instead of solving the complexities surrounding distributed data. Built on top of Spark, MLlib is a scalable machine learning library consisting of common learning algorithms and utilities, ...

(LR:Web14, databricks.com 03.02.2021)

Apache Spark Machine Learning Library (MLlib), ist dafür ausgelegt, hoch skalierbar sowie in Verbindung mit anderen Programmen und Datenquellen, Machine Learning einfach durchzuführen. Durch die hohe Geschwindigkeit, Skalierbarkeit und Kompatibilität zu anderen Frameworks, können diese Punkte durch Apache Spark bereitgestellt werden. Das nimmt den Programmierern bereits viel Arbeit ab.

Dadurch, dass es auf Apache Spark aufgebaut ist, kann man auch die Funktionen der anderen Modulen mitverwenden, so ist es möglich RDDs für die Datenaufbereitung und Reinigung zu verwenden und man kann auch durch die Spark Streaming API direkt mit Live-Daten trainieren und testen.

Funktionsweise

Auf die Funktionalität von Spark MLlib kann über zwei APIs zugegriffen werden:

- **RDD-based API**

Sollte man Spark MLlib mit RDD als Speicher im Hintergrund verwenden wollen, muss man dazu die Library `mllib` implementieren. Das ist die ursprüngliche API für das Arbeiten mit Machine Learning in Spark, das liegt daran, dass es in den Anfängen von Spark noch keine Dataframe-Integration gab.

Jedoch wurde mit Spark 2.0.0³¹ die RDD-based API in den Wartungsmodus umgestellt und die DataFrame-based API wurde zur primären Library erklärt.

- **DataFrame-based API**

Die DataFrame-based API ist über die Bibliothek `ml` erreichbar.

In der Dokumentation von Apache Spark MLlib werden drei Gründe für die Umstellung genannt:

- Die DataFrame-based API bietet im Gegensatz zur RDD-based API erhöhte Benutzerfreundlichkeit. Dabei liegen die Vorteile vor allem bei: Spark-Datenquellen, SQL/DataFrame-Abfragen, Tungsten- und Catalyst-Optimierungen
- DataFrames bieten eine einheitliche Methode über alle ML-Algorithmen hinweg und ermöglichen das Verwenden von anderen Sprachen.
- Das Arbeiten mit Pipelines³² wird durch DataFrames vereinfacht. Insbesondere Feature-Transformation.

(LR:Web15, vgl. spark.apache.org 05.02.2021)

Es ist auch noch möglich über Datasets Machine Learning zu betreiben, jedoch ist es sehr impraktikabel und es gibt daher auch keine eigene Library dafür.

Eine ausführliche Beschreibung worin der Unterschied bei der Verwendung von Apache Spark mit Datasets im Vergleich zu DataFrames oder RDD's liegt, findet man hier:
(LR:Article09, vgl. data-flair.training 09.02.2021)

Die Funktionsweise von Apache Spark MLlib setzt sich zusammen aus den unterschiedlichen Tools die verwendet werden:

1. ML Algorithmen
2. Featurization
3. Pipelines
4. Persistence
5. Utilities

³¹Erschienen 03.10.2016

³²siehe 6.2.3

1. Machine Learning Algorithmen

Apache Spark MLlib arbeitet mit vielen geläufigen learning Algorithmen, dabei wird in diesen Klassen unterteilt:

- Classification: Klassifizieren beschreibt das Einordnen von etwas in bestimmte Klassen, anhand von bestimmten Merkmalen (Features).

Dabei stehen zum Beispiel diese Algorithmen zur Verfügung:

- logistic regression
- naive Bayes
- ...

- Regression: Die Regressionsanalyse wird dazu verwendet um die Beziehung zwischen einem Merkmal zu einem oder mehreren anderen Merkmalen festzustellen. Somit können dann vorhersagen getroffen werden.

Dabei stehen zum Beispiel diese Algorithmen zur Verfügung:

- generalized linear regression
- survival regression
- ...

- Entscheidungsbaum: Ein Entscheidungsbaum wird verwendet um eine Reihe von Entscheidungen und ihrer Konsequenzen abzubilden. Er wird häufig im Zusammenhang mit Regression und Classification verwendet.

Unterstützte Methoden um mit Entscheidungsbäumen zu arbeiten sind:

- random forests
- gradient-boosted trees
- ...

- Recommendation System: Das Empfehlungssystem versucht vorherzusagen, welche Library oder Produkt ein Benutzer verwenden wird, und nimmt das als Grundlage für Empfehlungen für den Benutzer.

Dabei wird das Alternating least squares (ALS) verwendet. (LR:Article10, vgl. towardsdatascience.com 12.02.2021)

- Clustering: Beim Clustering geht es darum, Teilmengen von Entitäten mit ähnlichen Merkmalen, in einen Cluster zu vereinen. Es wird oft im Zusammenhang mit Pipelines und Entscheidungsbäumen verwendet.

In der Bibliothek `spark.mllib` werden folgende Modelle unterstützt:

- K-means
- Gaussian mixtures (GMMs)
- Latent Dirichlet allocation (LDA)
- Power iteration clustering (PIC)

(LR:Docs12, vgl. spark.apache.org 13.03.2021)

- Topic modeling: Mit Themenmodell können abstrakte Themen von einer Menge an Entities herausfinden. Dabei wird das Modell Latent Dirichlet allocation (LDA) unterstützt. LDA funktioniert ähnlich wie ein Clustering Algorithmus, somit gehört es zu beiden Klassen dazu.
- Weitere unterstützte Algorithmen sind:
 - Frequent itemsets
 - association rules
 - sequential pattern mining

(LR:Docs09, vgl. spark.apache.org 11.03.2021)

2. Featurization

Für das Arbeiten mit Features liefert MLlib vier Kategorien, mit vielen unterstützten Methoden:

- Feature Extractors: Herauslesen von Features aus Rohdaten.
Methoden dafür sind:
 - TF-IDF
 - Word2Vec
 - ...
- Feature Transformers: Skalieren, Abändern und Umwandeln von Features.
Methoden dafür sind:
 - Tokenizer
 - StopWordsRemover
 - n-gram
 - ...
- Feature Selectors: Ermöglicht das Arbeiten mit einer bestimmten Auswahl von Features.
Methoden dafür sind:
 - VectorSlicer
 - RFormula
 - ...
- Locality Sensitive Hashing (LSH): Verbindet Methoden der Featurization mit anderen ML Algorithmen.
Methoden dafür sind:
 - Approximate Nearest Neighbor Search
 - Bucketed Random Projection for Euclidean Distance
 - ...

Eine ausführliche Auflistung und Anleitung für jede Methode wird in der Dokumentation gegeben: (LR:Docs10, vgl. spark.apache.org 13.02.2021)

3. Pipelines

In this section, we introduce the concept of ML Pipelines. ML Pipelines provide a uniform set of high-level APIs built on top of DataFrames that help users create and tune practical machine learning pipelines.

(LR:Docs11, spark.apache.org 12.03.2021)

Bestandteile einer Pipeline:

- Transformer: Vereinfacht gesagt ist die Aufgabe eines Transformers, einen DataFrame in einen anderen umzuwandeln, meistens werden dabei an den ausgangs DataFrame Spalten angehängt. Dafür wird die Methode `.transform()` implementiert. Zum Beispiel:
 - Bei einem Feature-Transformer, wird eine Spalte gelesen und die Werte werden dann gemappt und an den DataFrame angehängt. Der DataFrame wird dann zurück geschickt.
 - Bei einem ML Algorithmus werden Features aus den DataFrame ausgelesen und bewertet. Anschließend wird der DataFrame um die Spalte 'Vorhersage' erweitert.
- Estimator: Ein Estimator bietet eine Abstraktion für einen Algorithmus, welcher mit Daten lernt. Dafür wird eine Methode `.fit()` implementiert, welches einen DataFrame verarbeitet und daraus ein Modell erstellt, das funktioniert intern wie ein Transformator. z. B.:
 - Eine LogisticRegression ist ein Estimator und wenn man nun mit `.fit()` ein DataFrame aufruft, wird ein Modell erstellt, welches wie ein Transformer funktioniert.
- Pipeline: Im normalen Betrieb ist es typisch das mehrere Pipelineschritte (Transformer + Estimator) hintereinanderlaufen, um von den Daten zu lernen und sie zu verarbeiten. Daher repräsentiert eine Pipeline eine Vielzahl von PipelineStages.
- Parameter: Estimator und Transformer haben jeweils eine Anzahl von Parametern. Dabei hat ein Parameter, ein `Param`, self-contained documentation³³ und kann auch als `ParamMap`(Schlüssel-Wert-Paare) dargestellt werden. z. B.:
 - Wenn man einen Estimator LogisticRegression hat, der einen Parameter `maxIter` hat. Könnte man `lr.setMaxIter(10)` verwenden, damit bei `.fit()` höchstens 10 Mal iteriert wird.
 - Wenn man zwei LogisticRegression Instanzen hat, kann man ihre Parameter über eine `ParamMap` festlegen: `ParamMap(lr1.maxIter -> 10, lr2.maxIter -> 20)`.

³³Jedem Wert wird eine eigene Caption zugeordnet

Wie funktioniert eine Pipeline?

Eine Pipeline ist einfache gesagt, eine Abfolge von unterschiedlichen Schritten, dabei stellt jeder Schritt entweder einen Estimator oder einen Transformer da. Diese werden in einer bestimmten Reihenfolge (meistens linear) abgearbeitet. Während es durch die Schritte durch geht, wird für jeden Transformer `.transform()` aufgerufen und für jeden Estimator die `.fit()`.

Ein Beispiel dafür wäre das Verarbeiten von einem einfachen Text mit einem 'Logistic Regression' Algorithmus, dafür werden drei Schritte benötigt:

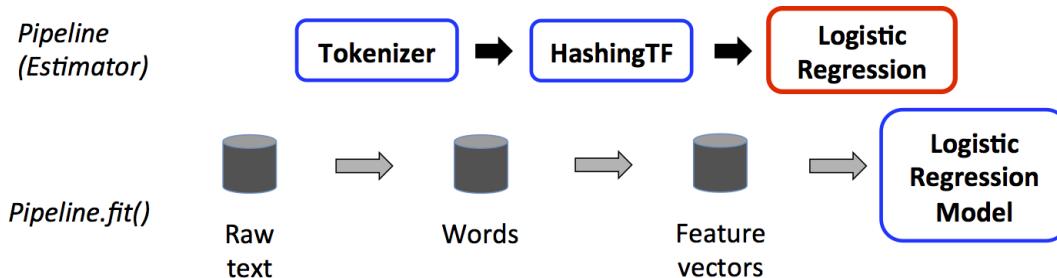


Abbildung 6.11: Logistic Regression Pipeline
Quelle: (LR:Docs11, vgl. spark.apache.org 13.03.2021)

Die erste Zeile stellt die verwendeten Methoden dar. Die ersten beiden Tokenizer und HashingTF sind Transformer (blau dargestellt), der letzte Schritt Logistic Regression (rot umrandet) ist ein Estimator. Die Zylinder in der unteren Reihe stellen die DataFrames dar.

Die Pipeline wird mit `.fit()` auf den Test aufgerufen. Als erster Schritt wird dann die `.transform()` vom 'Tokenizer' aufgerufen. Dieser teilt dann den Text in Wörter auf, somit wird der DataFrame um die Spalte Words erweitert. Als zweiter Schritt wird der Transformer 'HashingTF' wieder durch seine `.transform()` aufgerufen. Bei diesem Schritt werden aus den Wörtern Feature-Vektoren erstellt, diese Vektoren werden dann in einer neuen Spalte 'Feature vectors' abgespeichert. Der letzte Schritt ist der Estimator 'Logistic Regression', welcher durch die `.fit()` Methode einen neuen Transformer 'Logistic Regression Model' erstellt. Wäre der Estimator nicht der letzte Schritt, würde die `.transform()` vom neu-erstellten Transformer aufgerufen werden, bevor die Daten übergeben werden.

Vereinfacht kann man sagen, dass eine Pipeline sich wie ein Estimator verhält, es wird nämlich genauso, zuerst die `.fit()` Methode aufgerufen, welche dann ein Pipeline-modell erstellt und dieses Modell ist dann ein Transformer.

Das Pipelinemodell für dieses Beispiel würde in etwa so aussehen:

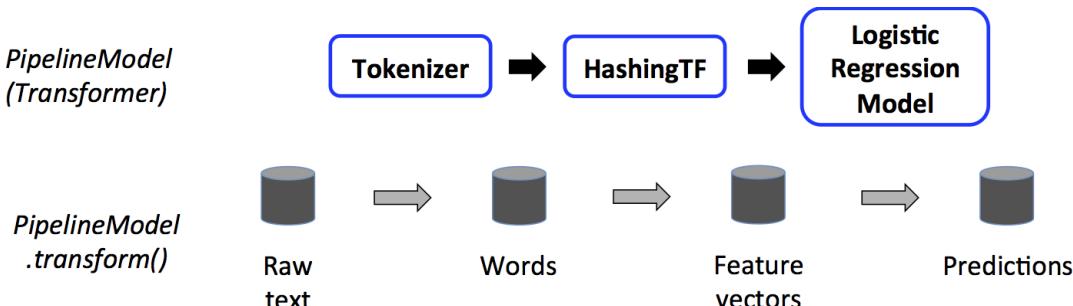


Abbildung 6.12: Logistic Regression Pipelinemodell
Quelle: (LR:Docs11, vgl. spark.apache.org 13.03.2021)

Die Schritte sind genau dieselben wie bei der oberen Abbildung von der Pipeline, das einzige, was sich geändert hat, ist, dass jetzt anstelle von den `.fit()` der Estimator, die `.transform()` von dem neu erstellten Transformer aufgerufen wird.

Dieses Pipelinemodell wird nun über den Befehl `.transform` auf einem Rohtext aufgerufen und der DataFrame wird dann durch die fitted (zu deutsch trainierte) Pipeline durchgeführt. Der große Vorteil davon ist, dass Testdaten immer durch dieselben trainierten Modelle verarbeitet werden.

(LR:Docs11, vgl. spark.apache.org 14.03.2021)

Einige Eigenheiten von Pipelines die man auf jeden Fall noch erwähnen muss:

- DAG Pipelines: Wie oben schon angekündigt gibt es neben linear ablaufenden Pipelines auch noch nicht-lineare Pipelines diese ist gültig solange ein DAG (Directed Acyclic Graph)³⁴ gebildet wird. Als Grundlage für die Abfolge werden die Namen der Eingabe- und Ausgabespalten verwendet. Eine solche Pipeline muss daher immer in topologischer Reihenfolge angegeben werden.
- Laufzeit Überprüfung: Dadurch das Pipelines mit DataFrames arbeiten, die unterschiedliche Typen verwenden, müssen diese überprüft werden, es ist jedoch nicht möglich das bereits beim Kompilieren des Programms zu machen. Daher werden bei Pipelines und Pipeline Modellen während der Laufzeit die Überprüfungen durchgeführt, bevor die Pipeline startet. Das geschieht durch ein sogenanntes DataFrame schema, welches für jede Pipeline die vorhergesagten DataFrames angibt.
- Einzigartige Pipeline Schritte: Ein jeder Schritt in einer Pipeline muss einzigartig sein das heißt, es kann nicht zweimal dieselbe Instanz von einem Transformer oder Estimator aufgerufen werden. Es ist trotzdem möglich einen Algorithmus zweimal durchzuführen, jedoch benötigt man dafür mehrere Instanzen. Das liegt daran, dass jeder Schritt durch einen eindeutige ID identifiziert wird.
- Stateless: Die `.transform` von einem Transformer und die `.fit()` von einem Estimator müssen immer stateless³⁵ sein. Es ist aber in zukünftigen Versionen geplant, dass auch stateful Algorithmen verwendet werden können.
- PreTrained Pipelines: Es gibt auch die Möglichkeit bereits vortrainierte (fitted) Pipelines herunterzuladen. Somit hat man sofort das Modell und kann direkt mit dem Testen beginnen.

Zwei Listen mit vielen verschiedenen PipelineModellen:

(LR:Web16, vgl. nlp.johnsnowlabs.com 12.02.2021)

(LR:GitHub02, vgl. gist.github.com 08.03.2021)

³⁴siehe 6.1.3

³⁵Bedeutung: Jede Anfrage wird gleich behandelt und es gibt keinen Zugriff auf vorhergehende Zugriffe oder Einstellungen

Beispiel einer Pipeline

Die vorher beschriebene Pipeline, schaut wie folgt aus, wenn man sie in Python umsetzt:

```

1 from pyspark.ml import Pipeline
2 from pyspark.ml.classification import LogisticRegression
3 from pyspark.ml.feature import HashingTF, Tokenizer
4
5 # Erstellen der Trainingsdaten
6 training = spark.createDataFrame(
7     [(0, "AI_Börse Projekt 5CHIF", 1.0),
8      (1, "1BHIF 2BHIF 3BHIF 4BHIF 5BHIF", 0.0),
9      (2, "Projektbetreuer Hasenzagl", 1.0),
10     (3, "Ersteklassen Zweiteklassen Dritteklassen Vierteklassen", 0.0),
11     (4, "Projektteam Raschbach Philipp Mader Asanov", 1.0),
12     (5, "1AHIF 2AHIF 3AHIF 4AHIF 5AHIF", 0.0)],
13     ["id", "text", "label"])
14 )
15 # instanziieren Der einzelnen Schritte
16 tokenizer = Tokenizer(inputCol="text", outputCol="words")
17 hashingTF = HashingTF(inputCol=tokenizer.getOutputCol(), outputCol="features")
18 lr = LogisticRegression(maxIter=10, regParam=0.001)
19
20 #Erstellen einer Pipeline aus den Schritten
21 pipeline = Pipeline(stages=[tokenizer, hashingTF, lr])
22
23 # Die Pipeliner mit den Daten trainieren
24 model = pipeline.fit(training)
25
26 # Erstellen von Testdaten, ohne Label!
27 test = spark.createDataFrame(
28     [(6, "AI_Börse Projekt 5CHIF"),
29      (7, "Ersteklassen 1AHIF 1BHIF"),
30      (8, "2AHIF macht Imagefilm über AI_Börse"),
31      (9, "Projektteam der 4AHIF in Zusammenarbeit mit der 4BHIF")],
32     ["id", "text"])
33 )
34
35 # Mit dem Pipelinemodele Testdaten verarbeiten
36 prediction = model.transform(test)
37
38 #Selektieren der wichtigen Spalten
39 selected = prediction.select("id", "text", "probability", "prediction")
40
41 #Ausgabe des labels
42 for row in selected.collect():
43     rid, text, prob, prediction = row
44     print("%d, %s --> prob=%s, prediction=%f" %(rid, text, str(prob),
45         prediction))

```

Listing 6.1: Example Pipeline

Der erste Schritt ist die Implementierung von den verwendeten Paketen. Dafür braucht man einmal den Import Pipeline und noch die drei Algorithmen: LogisticRegression, HashingTF und Tokenizer.

Anschließend werden Trainingsdaten erstellt. Dabei werden die Texte, welche in Verbindung mit dem Projekt *AI Börse* sind, mit dem Label 1 bewertet, die anderen mit dem Label 0.

Danach werden die einzelnen Algorithmen für die jeweiligen Schritte instanziert, bei den Transformern werden dafür die Spalte für die Eingabe `inputCol` und die Ausgabe `outputCol` angegeben. Bei dem Estimator werden Parameter übergeben, welche die Berechnungen beeinflussen.

Als letzter Schritt vor dem Trainieren werden die Transformer und der Estimator in einer gemeinsamen Pipeline gespeichert.

Dann wird die Pipeline mit dem Befehl `.fit()` mit den vorher erstellten Daten trainiert. Der nächste Schritt ist das Testen. Dafür werden wieder Daten erstellt, bei den Testdaten wird die Spalte `Label` nun ausgelassen, das liegt daran, dass diese Spalte vorhersagt werden soll.

Danach wird das Pipelinemodele, welches nun wie ein Transformer funktioniert mit dem Befehl `.transform()` auf den Testdaten aufgerufen. Der daraus resultierende DataFrame wird in `prediction` gespeichert.

Danach werden diese über eine For-Schleife ausgegeben. Die Ausgabe sieht dann so aus:

```
1 (6, AI_Börse Projekt 5CHIF) --> prob  
2   =[0.0018686541845726755, 0.9981313458154274], prediction=1.000000  
3 (7, Ersteklassen 1AHIF 1BHIF) --> prob  
4   =[0.983456813370943, 0.01654318662905699], prediction=0.000000  
5 (8, 2AHIF macht Imagefilm über AI_Börse) --> prob  
6   =[0.27614655892674644, 0.7238534410732536], prediction=1.000000  
7 (9, Projektteam der 4AHIF in Zusammenarbeit mit der 4BHIF) --> prob  
8   =[0.7549839089049043, 0.2450160910950957], prediction=0.000000
```

Listing 6.2: Example Pipeline Output1

Die Ausgabe setzt sich zusammen aus der ID des Textes und dem Text, dann mit einem Pfeil getrennt, wird die probability (zu deutsch Wahrscheinlichkeit) angegeben. Dabei steht der erste Wert für die Wahrscheinlichkeit das '0' das richtige Label ist und der zweite Wert dafür das es '1' ist. Die Prediction gibt, dann an für welchen Wert sich entschieden wurde. Das ist bei den ersten beiden Testdaten noch sehr eindeutig, jedoch ist es in den unteren zwei nicht mehr eindeutig.

(LR:Docs11, vgl. spark.apache.org 12.03.2021)

(LR:Web17, vgl. docs.databricks.com 06.03.2021)

4. Persistence

Das Persistence Tools bietet die Möglichkeit, das Algorithmen, Modelle oder Pipelines gespeichert und geladen werden können.

5. Utilities

'Utilities' (zu deutsch Hilfsmittel oder Werkzeuge) ist in Apache Spark MLlib ein sehr oft verwendeter Begriff. So gebrauchen ihn manche, um die Gesamtheit an Funktionalitäten von MLlib zu beschreiben, andere wiederum nur, um bestimmte Funktionen zusammen zufassen und dann gibt es auch noch die Libraries `pyspark.mllib.util.MLUtils` und `pyspark.ml.util`.

- Begriff als Sammlung von Funktionen:

In der offiziellen Dokumentation werden folgende Beispiele aufgezählt:

- Distributed linear algebra: Apache Spark MLlib verfügt über Algorithmen, die es für den Benutzer möglich machen, einfach eigene mathematische Ausdrücke verteilt auszuführen. Als Beispiel dafür werden Singular value decomposition (SVD) und Principal component analysis (PCA) genannt.
- Statistics: Apache Spark MLlib bietet auch Funktionen an, um Statistiken aus seinen Daten auszulesen. Die unterstützten Methoden dafür sind: Correlation, Hypothesis testing und Summarizer.
All diese Funktionen befinden sich in der Bibliothek `pyspark.ml.stat`.
In der Dokumentation werden auch die Methoden genauer beschrieben und es gibt ein Beispielcode für jede Methode.
- Data handling: Umfasst Methoden für das Arbeiten mit den Daten innerhalb von MLlib.

Somit bezieht sich diese Sichtweise von Utilities auf alle Funktionen von Apache Spark MLlib, die sonst keine genaue Zuordnung haben.

- Library `pyspark.mllib.util.MLUtils`

Wie man schon an dem `.mllib` erkennen kann, handelt es sich hierbei um eine RDD-based Library, also kann sie nicht in einer DataFrame-based API verwendet werden. Daher kann man sagen, dass es sich bei der Interpretation von Utilities um eine Ältere handelt, die nur in RDD-based APIs auftritt.

Die Library an sich bietet Hilfsmethoden an, um die verwendeten Daten in MLlib zu laden und zu speichern, außerdem liefert sie auch schon pre-processed Data (zu deutsch vorverarbeitete/vorbereitete Daten)

- Library `pyspark.ml.util` Die Library `pyspark.ml.util` liefert ähnlich wie die `pyspark.mllib.util.MLUtils` Methoden, um MLReader, MLWriter und ML-Instanzen zu erstellen, zu speichern, zu laden und zu bearbeiten.

Kompatibilität mit anderen Programmen

Man kann Apache Spark MLlib überall dort verwenden, wo man eine Apache Spark Instanz laufen lassen kann. So ist es möglich, MLlib auf einem eigenständigen Single-node zu verwenden, sowie auf einem großen Computersystem welches über EC2³⁶, Hadoop YARN, Mesos oder Kubernetes³⁷ verwaltet wird.

Außerdem bietet Apache Spark MLlib auch unzählige Schnittstellen für das Erlangen von Daten, so ist es möglich, auf Daten von zum Beispiel HDFS, Apache Cassandra, Apache HBase oder Apache Hive zuzugreifen. Des Weiteren unterstützt die Library auch die unterschiedlichsten Datentypen, so ist es möglich, neben den gängigen Files (Parquet, CSV, JSON and JDBC), auch Bilder und LIBSVM³⁸ Dateien damit zu verarbeiten.

Auf die Apache Spark MLlib API kann über 4 Programmiersprachen zugegriffen werden: Java, Scala, Python, R

GraphX (graph)

GraphX fasst mehrere Funktionen in ein System zusammen. Hauptbestandteil davon ist ETL³⁹, Graphverarbeitung und explorative Analyse innerhalb der Daten. Dafür werden mehrere Algorithmen verwendet. Jedoch ist die Analyse von Daten in einem Cluster durch Graphen nicht Bestandteil dieser Arbeit, daher wird nicht näher darauf eingegangen.

(LR:Docs15, vgl. spark.apache.org 14.03.2021)

³⁶Amazon Elastic Computer Cloud

³⁷siehe 6.3.1

³⁸Format zum Abspeichern von Feature Vektoren

³⁹Extract, transform, load

6.2.4 Pythonic

Zum 10-jährigen Jubiläum⁴⁰ vom ersten Spark Release wurde die Version 3.0 veröffentlicht. Zu diesem Anlass hat das Unternehmen Databricks, die Anzahl von allen ausgeführten Kommandos auf ihrem System pro API in dem Jahr 2013⁴¹ mit 2020 verglichen:

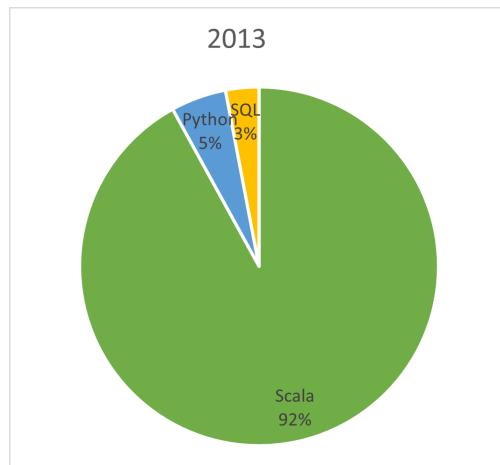


Abbildung 6.13: Pythonic Graph 2013
Quelle (LR:Video01, vgl. databricks.com 12.03.2021)

Man kann erkennen, dass im Jahre 2013 fast ausschließlich die Spark Scala API im Einsatz war. Das liegt daran, dass Scala die 'first-class API' von Spark ist, was so viel bedeutet wie, dass alles darauf ausgelegt ist mit dieser API zu funktionieren.

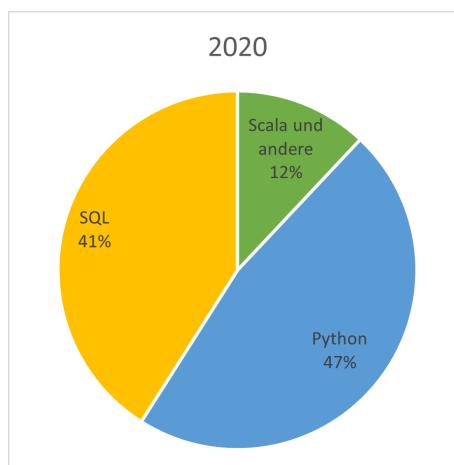


Abbildung 6.14: Pythonic Graph 2020
Quelle (LR:Video01, vgl. databricks.com 12.03.2021)

⁴⁰Juni 2020

⁴¹Gründungsjahr von Databricks

Heutzutage wird Apache Spark immer mehr als gesamtes System verwendet, welches eine große Menge von unterschiedlichen Aufgaben erfüllt. Zum Beispiel Data Analytics, BI, Streaming, Data Science und Machine Learning. Somit benutzen jetzt nicht mehr nur 'Data Engineers', welche für das Erstellen ihrer rechenaufwendigen Aufgaben Scala API benötigen, Spark, sondern auch 'Data Scientists', die Python API verwenden und 'Data Analysts', welche die SQL API bevorzugen.

Project Zen

Somit hat Databricks das Projekt 'Project Zen'⁴² ins Leben gerufen. Ziel dieses Projektes ist es, Apache Spark mehr 'Pythonic' zu gestalten. Pythonic bedeutet, dass das Arbeiten mit der Phyton API vereinfacht wird und somit benutzerfreundlicher ist.

Dabei werden folgende Punkte überarbeitet:

- User Defined Functions (UDF): In der alten Python API musste bei jeder benutzer-generierten Funktion, der Typ der verwendeten Variablen innerhalb eines Tags angegeben werden.
- Error Messages: Die alte Python API war dafür bekannt, sehr lange nichts aussagende Fehlermeldungen auszugeben. Auch dieses Problem soll mit der überarbeiteten API Version behoben werden und somit soll eine bis zu sechs Seiten lange Ausgabe auf eine Seite reduziert werden.
- Type Pins: Es ist in der neuen API möglich, über ein ganzes Projekt hinweg, den Datentyp von Variablen statisch zu überprüfen.
- Auto Complete: Mit 'Project Zen' wird eine neue Autovervollständigung mitgeliefert, welche den Kontext verstehen kann und somit die meist relevanten Antworten liefert.
- Docs: Es soll im Laufe des Projektes auch die Dokumentation von Spark Python API überarbeitet werden. Dabei wurde sich an den NumPy Style von der Dokumentation orientiert. Auf der neuen Hauptseite stehen nun alle Pakete von Spark und jede Klasse wird mit einem Satz beschrieben. Sollte man nun auf eine Klasse drücken, wird eine genauere Beschreibung der Klasse und eine Liste von allen Methoden angezeigt. Auf diese Methoden kann man dann wieder drücken, um auf eine eigene Seite für die Methode zu gelangen.

(LR:Video01, vgl. Project Zen: Making Spark Pythonic 12.03.2021)

⁴²Anfang 2020

Koalas

Mit Koalas ist es möglich, die DataFrame API von Pandas in Apache Spark zu verwenden. Pandas ist der 'state of the art' für das Arbeiten mit DataFrames auf einem Singenode-System in Python, währenddessen Apache Spark der Standard für das Verarbeiten von Big Data ist. Mit Koalas kann man die beiden Frameworks vereinen. Dabei bietet es folgende Vorteile:

- Man kann sein derzeitiges Programm direkt auf Apache Spark hoch skalieren, ohne sich mit Spark zu beschäftigen, solange man sich mit Pandas auskennt.
- Man kann über ein Programm sowohl Pandas als auch Spark verarbeiten.

(LR:Docs13, vgl. koalas.readthedocs.io 13.03.2021)

6.2.5 Fazit

Vor- und Nachteile von Apache Spark

- **Vorteile**

Aus dem bis jetzt vorgestellten Informationen kann man einige Vorteile erkennen:

1. Geschwindigkeit: Wie bereits erwähnt, kann Apache Spark bis zu 100-mal schneller BigData verarbeiten, als Hadoop. Das liegt daran, dass Spark in-Memory Speicher verwendet, währenddessen benutzt Hadoop lokale Speicher. Auch im Vergleich zu anderen Frameworks wie Pandas, ist es schneller. Mit dem Spark ist es möglich mehrere Petabytes (PB⁴³) an Daten, auf mehrere Tausend Nodes zu verarbeiten.

(LR:Article11, vgl. databricks.com 07.03.2021)

2. Benutzerfreundlichkeit: Apache Spark liefert verschiedene APIs, somit kann man in seiner bevorzugten Sprache programmieren. Somit ist es möglich mit den einfach zu verstehende APIs große Daten zu verarbeiten.
3. Analytik: Es bietet viele Möglichkeiten, um seine Daten zu analysieren (z. B.: Machine Learning, Graphen Analyse, Streaming, MAP, ...).
4. Dynamische Natur: Man kann parallel arbeitende Prozesse entwickeln.
5. Multilingual: Wie schon angesprochen, ist es möglich Apache Spark über unterschiedliche Sprachen zu bedienen.
6. Leistungsstark: Apache Spark ist in der Lage, viele BigData Aufgaben gleichzeitig durchzuführen, das ist durch die in-Memory Datenverarbeitung.

⁴³1PB = 1024Terabytes (TB); 1TB = 1024Gigabyte (GB)

7. Kompatibilität: Die bereits angesprochene Kompatibilität von Apache Spark ist einer der wichtigsten Punkte von Spark.
8. Open-Source: Die große Community, die für die Entwicklung von Apache Spark verantwortlich ist, ist auch ein großer Vorteil von Apache Spark. Denn somit stehen viele Beispielprogramme zur Verfügung und dadurch ist auch die Entwicklung breit gefächert.

- **Nachteile**

Neben seinen vielen Vorteilen, besitzt das Framework jedoch auch ein paar Nachteile. Beispiele dafür sind:

1. Keine automatische Optimierung: Apache Spark bietet keine Methoden zum Optimieren des Codes, somit muss jeder Code selbst optimiert werden. Das könnte zu Problemen führen, wenn viele anderen Programme diese Funktion anbieten.
2. Dateiverwaltung: Apache Spark hat kein eigenes Filesystem. Somit braucht man immer ein Programm, auf dem Spark aufgesetzt werden kann, welches sich um die Datenverwaltung kümmert. Meistens wird dafür Hadoop oder ein Cloud-basierter Service herangezogen.
3. Algorithmen: Es gibt Libraries, die eine größere Anzahl von Machine Learning Algorithmen und Funktionen anbietet, als Apache Spark MLlib.
4. Kleine Dateien: Sollte man Spark über Apache Hadoop verwenden, kann man auf ein Problem stoßen, wenn man zu viele kleine Daten verarbeiten will. Das Problem hat seinen Ursprung darin, dass Hadoop lieber eine kleine Anzahl von großen Daten, als eine große Anzahl von kleinen Daten verarbeitet. Verschlimmern kann sich das Problem, wenn die Dateien komprimiert sind, denn so muss die gesamte Datei geladen werden, bevor man mit dem Entpacken beginnen kann. Das kann zur Überlastung des gesamten Systems führen. In weitere Folge wird dann für jedes File eine Partition im RDD erstellt, was zu Millionen kleinen Partitionen führen kann.
5. Window Criteria: Apache Spark verwendet nur zeitgesteuerte Bedingungen (Criteria). Somit ist es nicht möglich, eine Bedingung Eintrags-orientiert (record-based) durchzuführen. Der Unterschied ist, dass beim record-based window criteria immer ein Eintrag zu einem Zeitpunkt verarbeitet wird. Während dessen werden in Spark immer kleine Batches zu einem Zeitpunkt verarbeitet.
6. Mehrbenutzerbetrieb: Es gibt keine Lösung für konkurrierende Zugriffe, deshalb ist ein Mehrbenutzerbetrieb nicht möglich.

Optimale Einsatzbereiche

Apache Spark kann optimal als zentrale ETL (Extract, Transform, Load) Engine eingesetzt werden, zum Beispiel für einen Data Lake (eine große Menge an 'rohen' Daten, die lose gespeichert werden). Der große Vorteil von Spark ist, dass es in der Lage ist, täglich riesige Datenmengen zu verarbeiten (bewegen, filtern und transformieren). Spark eignet sich außerdem so gut, weil damit viele unterschiedliche Datentypen in den Data Lake gespeichert werden können und es sehr flexibel ist. Viele bestehenden Datenbanken sind nur auf eine bestimmte Anzahl begrenzt und daher kann sich Apache Spark in diesem Bereich hervorheben.

Eine Architektur, in der Apache Spark als ETL Engine dafür verwendet wird, um ein Datawarehouse (DWH) und ein Datalake zu verwalten, könnte so aussehen:

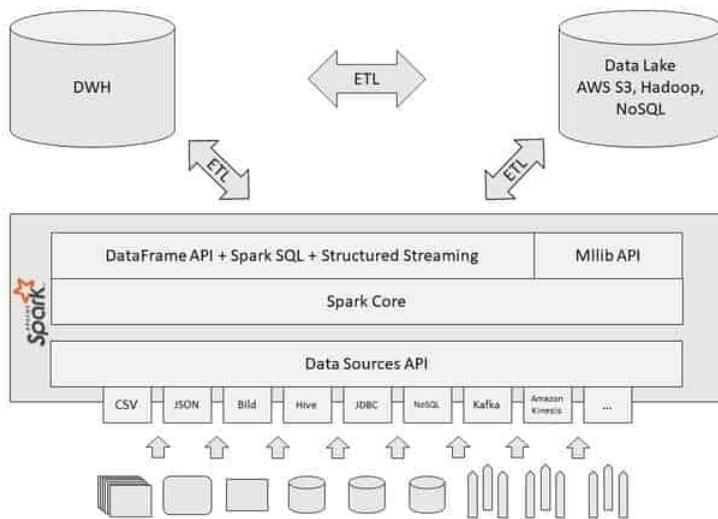


Abbildung 6.15: Spark als ETL Engine
Quelle: (LR:Article12, vgl. datasolut.com 13.03.2021)

Durch Apache Spark ist es möglich, die rohen und unformatierten Daten aus dem Data Lake, in das DWH zu speichern und umgekehrt. Des Weiteren bietet Spark mit der Data Sources API eine Schnittstelle, für jegliche Arten von Daten, welche wiederum in das DWH oder in den Data Lake gespeichert werden können.

(LR:Article12, vgl. datasolut.com 14.03.2021)

Weitere Einsatzbereiche von Spark sind überall wo:

1. große Datenmengen analysiert werden sollen,
2. Streaming von Echtzeitabfragen verlangt wird und
3. Machine Learning durchgeführt werden soll.

Apache Spark wird auch von vielen großen Firmen verwendet. Beispiele dafür sind: Netflix, MyFitnessPal, eBay, Yahoo News und viele mehr.
(LR:Article12, vgl. datasolut.com 14.03.2021)

Abschließende Wörter

Abschließend kann man sagen, dass Apache Spark der Spitzenreiter, im Arbeiten mit BigData ist, da es so vielfältig ist und nahezu alles damit möglich ist. Außerdem ist es auch noch eines der am schnellsten arbeitenden Frameworks (wenn nicht sogar das schnellste). Daher wird man Apache Spark in der Zukunft immer häufiger sehen.

6.3 Weitere Implementationen

Abgesehen von Apache Spark und Apache Hadoop, die meistens in Verbindung gebracht werden wenn es um das Arbeiten mit Cluster geht, gibt es noch mehrere kleinere Frameworks, die genauso Methoden zum skalieren benutzen, damit sie einen Cluster verwalten können.

6.3.1 Kubernetes

Kubernetes is an open source container orchestration engine for automating deployment, scaling, and management of containerized applications. The open source project is hosted by the Cloud Native Computing Foundation (CNCF)

(LR:Docs14, kubernetes.io 14.03.2021)

Kubernetes(meist abgekürzt mit K8s) ist eine Orchestrierung, die das Arbeiten mit Container automatisiert.

K8s arbeitet auch mit Cluster, dabei gibt es Node Geräte, welche Container-Applikationen abarbeiten und ein Managementebene, welche für das Verwalten der Nodes zuständig ist.

Die Funktion des Clustering ist sogar eines der Hauptvorteile von Kubernetes, weil so können Container über mehrere Systeme hinweg ausgeführt werden und sie sind nicht an einzelne Maschinen gebunden.

(LR:Web22, vgl. redhat.com 14.03.2021)

6.3.2 Mesos

Apache Mesos ist genau so wie Hadoop YARN ein scheduler (zu deutsch Planer). Diese sind darauf ausgelegt, die Workload innerhalb eines Clusters auf die unterschiedlichen Nodes aufzuteilen. Dabei bietet Mesos die Möglichkeit an, alle Ressourcen, die in einem Data Center auftreten, zu bewältigen. Das beschreibt auch den Unterschied zwischen Mesos und YARN, denn YARN ist nur in der Lage, Hadoop Jobs zu verwalten und aufzuteilen.

Somit bietet sich Mesos perfekt für die Skalierung von Non-Hadoop Frameworks auf ein Clustersystem und YARN für die Skalierung von Hadoop auf mehreren Tausend Cluster an.

Es gibt auch ein eigens Framework, Myriad, welches es ermöglicht YARN über Mesos zu bedienen. Durch dieses Framework ist es möglich, durch in einem Mesos System auch Hadoop optimierte Ressourcenanfrage durch YARN zu verwalten.

Man kann auch auf einen Cluster, der mit Mesos verwaltet wird, Apache Spark laufen lassen.

(LR:Article14, vgl. oreilly.com 14.03.2021)

6.4 Alternativen

Es gibt auch in der heutigen Zeit noch Methoden und Frameworks, die es schaffen BigData zu verarbeiten ohne direkt einen Cluster zu verwenden.

6.4.1 Dremel

Dremel is a scalable, interactive ad-hoc query system for analysis of read-only nested data. By combining multi-level execution trees and columnar data layout, it is capable of running aggregation queries over trillion-row tables in seconds.

(LR:Web21, research.google 14.03.2021)

Dremel ist ein Datenanalyse-Werkzeug, welches in der Lage ist, SQL-Type-Abfragen auf große, strukturierte Datasets durchzuführen (z. B.: Log Datein). Es benutzt zwar eine SQL ähnliche Syntax, jedoch ist es nicht in der Lage ein 'Create' oder 'Alter'-Befehl durchzuführen und ist somit nur readonly. Dremel ist dabei hoch skalierbar und vor

allem auf ad-hoc⁴⁴ Abfragen ausgelegt.

Es funktioniert, indem es die SQL-Abfragen in 'execution trees' (ähnlich wie 'Decision Trees', nur mit ausführbaren Aufgaben statt Entscheidungen) darstellt und diese abarbeitet.

Beziehung zu Hadoop

Apache Hive, hat eine ähnlichen Anwendungsbereich wie Dremel, denn mit Hive ist es auch möglich, über SQL-Syntax Abfragen auf Hadoop zu machen. Nur das bei Apache Hive, die SQL Abfrage in eine MapReduce Methode umgeschrieben werden.

MapReduce ist jedoch nicht darauf ausgelegt, als Analysetool verwendet zu werden, sondern für das Arbeiten mit ClusterComputer. Deshalb hat Apache Hive eine sehr hohe Latency⁴⁵.

Googles Dremel versucht dabei aber nicht Hadoop zu verdrängen. Es ist eher eine Erweiterung von Hadoop, die es möglich macht, Analyse-Aufgaben über eine SQL-Syntax durchzuführen, ohne der erhöhten Latency, die man bei Apache Hive hat. Das ist wegen dem MapReduce Algorithmus im Hintergrund.

(LR:Web20, vgl. intellipaat.com 14.03.2021)

Teil von BigQuery

Dremel ist genauso wie Colossus, Jupiter und Borg die Technologie hinter Googles BigQuery Framework. BigQuery ist ein DWH, welches ohne einen Server funktioniert. Seine Hauptaufgabe ist das Analysieren von Daten.

Dremel wird als die Abfrage-Engine in BigQuery verwendet.

(LR:Web19, vgl. cloud.google.com 14.03.2021)

⁴⁴schnelle

⁴⁵Verzögerung

Kapitel 7

Implementation von Clustersystemen im Projekt AI Börse

7.1 Installation von PySpark auf einem Windows Server

Damit PySpark auf dem Windows Server richtig funktioniert, ist es wichtig, dass man zuerst Apache Spark, Java und Hadoop auf dem Server aufsetzt. Weiters muss dann natürlich auch Python installiert sein.

Ein sehr wichtiger Punkt, um Fehler zu vermeiden, ist es, darauf zu achten, untereinander kompatible Versionen zu verwenden. Daher wird für den Windowserver von der Applikation AlBörse folgende Versionen installiert:

Software	verwendete Version
JDK	1.8.0_275
Apache Spark	2.4.4
Apache Hadoop	2.7
Python	3.6.8
Pyspark	2.4.4

Tabelle 7.1: Verwendete Versionen für die PySpark Implementierung

Setup JDK

Damit Apache Spark und auch PySpark funktioniert, wird eine JDK am System vorausgesetzt, diese kann einfach herunterladen werden.

Auf dem Server wird Java im Verzeichnis C:\java installiert, damit alle Programme Zugriff haben.

Überprüfen, ob die Installation auch erfolgreich war, kann man über den Befehl. `java -version`:

```
1 C:\Users\20160584>java -version
2 openjdk version "1.8.0_275"
3 OpenJDK Runtime Environment (AdoptOpenJDK) (build 1.8.0_275-b01)
4 OpenJDK 64-Bit Server VM (AdoptOpenJDK) (build 25.275-b01, mixed mode)
```

Listing 7.1: Überprüfung der Java Version auf dem Server

Setup Hadoop

Man muss keine ganze Hadoop Instanz installieren. Es wird für die Implementierung von PySpark nur die `winutils.exe` benötigt.

Diese kann man sich über das GitHub Verzeichnis (LR:GitHub01, vgl. [github.com 30.12.2020](https://github.com/30.12.2020)) installieren.

Abgelegt haben wurde es im Verzeichnis C:\hadoop\bin.

Setup Apache Spark

Um Apache Spark auf dem Server aufzusetzen, muss man zuerst auf die Website gehen und über das (LR:Web04, vgl. archive.apache.org 31.12.2020) die richtige Version auswählen. Im Falle des Servers wird die Version spark-2.4.4 ausgewählt. Danach erscheint eine Liste mit verschiedenen .tgz¹ Dateien.

¹TAR-Archivdatei, die mit der Software Gzip komprimiert wurde

Index of /dist/spark/spark-2.4.4

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 Parent Directory		-	
 SparkR_2.4.4.tar.gz	2019-08-27 22:01	310K	
 SparkR_2.4.4.tar.gz.asc	2019-08-27 22:01	819	
 SparkR_2.4.4.tar.gz.sha512	2019-08-27 22:01	207	
 pyspark-2.4.4.tar.gz	2019-08-27 22:01	206M	
 pyspark-2.4.4.tar.gz.asc	2019-08-27 22:01	819	
 pyspark-2.4.4.tar.gz.sha512	2019-08-27 22:01	210	
 spark-2.4.4-bin-hadoop2.6.tgz	2019-08-27 22:01	218M	
 spark-2.4.4-bin-hadoop2.6.tgz.asc	2019-08-27 22:01	819	
 spark-2.4.4-bin-hadoop2.6.tgz.sha512	2019-08-27 22:01	268	
 spark-2.4.4-bin-hadoop2.7.tgz	2019-08-27 22:01	219M	
 spark-2.4.4-bin-hadoop2.7.tgz.asc	2019-08-27 22:01	819	
 spark-2.4.4-bin-hadoop2.7.tgz.sha512	2019-08-27 22:01	268	
 spark-2.4.4-bin-without-hadoop-scala-2.12.tgz	2019-08-27 22:01	137M	
 spark-2.4.4-bin-without-hadoop-scala-2.12.tgz.asc	2019-08-27 22:01	819	
 spark-2.4.4-bin-without-hadoop-scala-2.12.tgz.sha512	2019-08-27 22:01	193	
 spark-2.4.4-bin-without-hadoop.tgz	2019-08-27 22:01	158M	
 spark-2.4.4-bin-without-hadoop.tgz.asc	2019-08-27 22:01	819	
 spark-2.4.4-bin-without-hadoop.tgz.sha512	2019-08-27 22:01	288	
 spark-2.4.4.tgz	2019-08-27 22:01	15M	
 spark-2.4.4.tgz.asc	2019-08-27 22:01	819	
 spark-2.4.4.tgz.sha512	2019-08-27 22:01	195	

Abbildung 7.1: Unterschiedlichen Version von Apache Spark
Quelle: (LR:Web04, vgl. archive.apache.org 31.12.2020)

Der größte Unterschied, der unterschiedlichen Versionen, ist ihr Verhältnis zu Apache Hadoop. Es gibt welche, die sind auf Hadoop ausgelegt², dann gibt es welche, die frei gelöst von Hadoop sind und welche für die Programmiersprachen R³ und Python⁴. Auf dem Server wurde die Version spark-2.4.4-bin-hadoop2.7.tgz heruntergeladen, da die Winutils von der selben Version ist.

Anschließend wurde das Paket im Verzeichnis C:\Spark abgelegt.

²Ihr für die Versionen 2.6 und 2.7

³SparkR

⁴pyspark

Überprüfen der Apache Spark Version

In der Liste der unterschiedlichen Apache Spark Versionen gibt es neben der eigentlichen .tgz Datei auch immer eine mit dem Kürzel .asc⁵ oder .sha512⁶ angehängt. Das sind die jeweiligen Dateien verhasht. Das kann dazu genutzt werden, um die eigene Version auf ihre Integrität zu überprüfen.

Wenn man nun mit dem Befehl `certutil -hashfile` seine lokale Version verhasht, kann man die beiden Werte vergleichen.

```
1 C:\Users\20160584>certutil -hashfile C:\Users\20160584\Downloads\spark
2   -2.4.4-bin-hadoop2.7.tgz SHA512
3   SHA512-Hash von C:\Users\20160584\Downloads\spark-2.4.4-bin-hadoop2.7.tgz
4   2e3a5c853b9f28c7d4525c0adcb0d971b73ad47d5cce138c85335b9f53a6519540d3923
5   cb0b5cee41e386e49ae8a409a51ab7194ba11a254e037a848d0c4a9e5
5 CertUtil: -hashfile-Befehl wurde erfolgreich ausgeführt.
```

Listing 7.2: Verhashen von Apache Spark



Abbildung 7.2: Hash einer Apache Spark Version
Quelle: (LR:Web04, vgl. archive.apache.org 31.12.2020)

Somit kann sichergestellt werden, dass die Version, welche auf den Server geladen ist, nicht korrupt ist.

Einstellen von Umgebungsvariablen

Damit nun alle Dateien auf Spark zugreifen können, muss man noch Umgebungsvariablen darauf legen.

Dazu braucht man:

- HADOOP_HOME auf C:\hadoop

⁵ASCII-Datein

⁶vom Englischen secure hash algorithm

- SPARK_HOME auf C:\spark
- JAVA_HOME auf C:\java
- Path muss noch um %HADOOP_HOME%\bin und %SPARK_HOME%\bin erweitert werden.

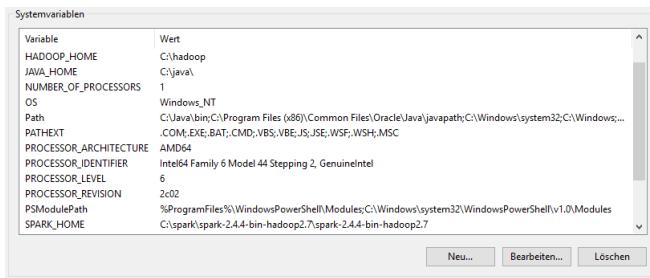


Abbildung 7.3: Anzeige der HOME Variablen

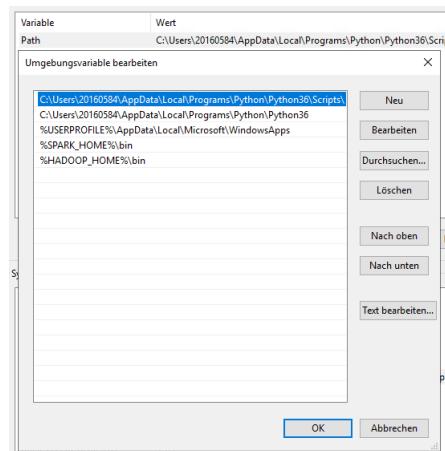


Abbildung 7.4: Anzeige der Path Variable

Einstellen von Rechten

Schließlich muss man noch Rechte für die C:\tmp Datei festlegen.
Durch die Winutils geht das ganz einfach mit dem Befehl chmod

```
1 C:\Users\20160584>%HADOOP_HOME%\bin\winutils.exe chmod 777 /tmp/hive
2 C:\Users\20160584>%HADOOP_HOME%\bin\winutils.exe chmod 777 /tmp/
```

Listing 7.3: Festlegen von Rechten von \tmp

PySpark

Nun kann PySpark über den Python Befehl `python3 -m pip install pyspark==2.4.4` heruntergeladen werden.

Da jetzt alles funktioniert, ist es durch den Befehl `PySpark` möglich, in der Eingabeaufforderung eine neue PySpark Instanz zu starten.

```
1 C:\Users\20160584>PySpark
2 Python 3.6.8 (tags/v3.6.8:3c6b436a57, Dec 24 2018, 00:16:47) [MSC v.1916 64
   bit (AMD64)] on win32
3 Type "help", "copyright", "credits" or "license" for more information.
4 Using Spark's default log4j profile: org/apache/spark/log4j-defaults.
   properties
5 Welcome to
6
7   / _/ \
8   \ \_ \_ \_ \_ \_ \_ \_ \_ \_ \
9   /_ / . / \_, /_ / /_ / \_ \_ \_ \_ \
10  /_ /
11
12 Using Python version 3.6.8 (tags/v3.6.8:3c6b436a57, Dec 24 2018 00:16:47)
13 SparkSession available as 'spark'.
14 >>>
```

Listing 7.4: PySpark starten

Zwei gute Anleitungen zur Installation von PySpark werden auf den beiden Seiten (LR:Web05, stackoverflow.com 1.1.2021) und (LR:Web06, phoenixnap.com 1.1.2021) aufgezeigt.

7.2 Einbindung von PySpark in die Analyse von Börsenbriefe

7.2.1 Pipelines

Die Programme, die bei diesem Ansatz verwendet wurden, sind auf dem Datenträger unter X:\Programme\sentiment_analyse zu finden.

In dem Projekt AI Börse wurde entschieden, dass Apache Spark über vortrainierte Pipelines eingebunden wird.

Der Teil des Programms, welches ursprünglich zur Analyse von Börsenbriefen verwendet werden sollte und welcher Pipelines nutzt, sieht so aus:

```
1  from pyspark.ml import Pipeline
2  from pyspark.sql import SparkSession
3  import pyspark.sql.functions as F
4  from sparknlp.annotator import *
5  from sparknlp.base import *
6  import sparknlp
7  from sparknlp.pretrained import PretrainedPipeline
8
9  import os
10 os.environ["PYSPARK_PYTHON"] = "/usr/bin/python3.6"
11
12 spark = sparknlp.start()
13 print(sparknlp.version())
14 print(spark.version)
15 ...
16
17 pipeline = PretrainedPipeline("analyze_sentiment", lang = "en")
18 result = pipeline.transform(df_page_contents_english)
19 df_result = result.select("text", "sentiment").toPandas()
20 df_sentences["sentiment0"] = df_result["sentiment"]
21 print(df_sentences)
22
23 for index, row in df_sentences.iterrows():
24     row_data = row["sentiment0"][0]
25     if row_data.result == "positive":
26         row["sentiment0"] = row_data.metadata["confidence"]
27     elif row_data.result == "negative":
28         row["sentiment0"] = str(-1 * float(row_data.metadata["confidence"]))
29     else:
30         row["sentiment0"] = 0
31
32 print(df_sentences)
```

Listing 7.5: Einbindung von Pipelines

Erklärung des Codeabschnittes

1. Imports

Zuerst werden die notwendigen Pakete für PySpark importiert:

- `from pyspark.ml import Pipeline`
Das Paket `pyspark.ml` beinhaltet, die allgemeinen Funktionen von der Apache Spark MLlib. Es verwendet im Hintergrund DataFrames zum Verarbeiten der Daten. In diesem Codeabschnitt wird nur die Funktionalität einer Pipeline implementiert.
- `from pyspark.sql import SparkSession`
Das Paket `pyspark.sql` ermöglicht, denn Zugriff auf die Funktionen von Spark SQL über Python. Somit kann man SQL Befehle auf einer Datenbank ausführen. Es wird jedoch nur die Funktion `SparkSession` integriert. Mit dieser Funktion kann man eine Session starten.
- `import pyspark.sql.functions as F`
Die Library `pyspark.sql.functions` ist eine Liste von mehreren built-in⁷ Funktionen für das Arbeiten mit DataFrames. Die Funktionalität ist über den Stellvertreter `F` erreichbar.
- `from sparknlp.annotator import *`
Bei diesem Import werden die Funktionen eines *SparkNLP Annotator*⁸ eingebunden.
- `from sparknlp.base import *`
Mit dem Import `*` von `sparknlp.base` werden alle Basisfunktionen von *SparkNLP* implementiert.
- `import sparknlp`
Das Paket `sparknlp` muss auch noch implementiert werden, damit auf die Standardfunktionen von *SparkNLP* zugegriffen werden kann.
- `from sparknlp.pretrained import PretrainedPipeline`
Abschließend werden noch die pretrained Pipelines von *SparkNLP* über das Paket `pretrained` in das Programm eingebunden.

2. Setzen der Umgebungsvariable

Als nächstes muss noch die Umgebungsvariable `PYSPARK_PYTHON` auf das Pythonverzeichnis gesetzt werden, weil in diesem Verzeichnis die Library Pyspark liegt, welche über pip installiert wurde. Es sollte auch darauf geachtet werden, dass die richtige Version angegeben wird, damit es nicht zu Komplikationen zwischen zwei oder mehreren unterschiedlichen Versionen kommt.

⁷zu deutsch integrierter

⁸Befehle zum anzeigen von Zusatzinformationen

3. Starten der Session

Anschließend muss eine Spark Session gestartet werden. Das erfolgt über den Befehl `sparknlp.start()`. Dabei wird eine automatische Session mit Standardattribute gestartet. Will man die Attribute genauer angeben, kann der Befehl `SparkSession.builder` verwendet werden. Dieser funktioniert über einen Dekoriere⁹ und kann so unterschiedliche Konfigurationen übernehmen.

Bei diesem Code werden danach auch noch, zum Überprüfen, die Versionen ausgegeben:

```
1 2.7.5  
2 2.4.4
```

Listing 7.6: Output praxis Pipeline1

4. Benutzung der Pipeline

Der letzte Schritt ist das Verwenden einer konkreten Pipeline.

Dabei wird zuerst die vortrainierte Pipeline `analyse_sentiment` instanziert, dabei muss auch die Variable `lang` gesetzt werden, mit dieser wird die Sprache des Modells angegeben.

Danach wird die Pipeline wie ein Modell¹⁰ über den Befehl `.transform()` auf ein Pandas DataFrame aufgerufen. Jetzt durchläuft der DataFrame die einzelnen Schritte der Pipeline und wird anschließend in den Spark DataFrame `result` gespeichert. Für die Weiterverarbeitung wird aus dem Spark DataFrame nur die beiden Spalten `text` und `sentiment` in den DataFrame `df_result` gespeichert.

Als nächster Schritt, wird das Sentiment in ein übergeordneten DataFrame gespeichert. In einer folgenden For-Schleife wird dann überprüft, ob die Vorhersage negativ oder positiv ist, je nachdem wird der `confidence`-Wert¹¹, als positiver oder negativer Wert abgespeichert.

Zur Überprüfung wird der Dataframe zwischendurch ausgegeben.

Das Programm wird so jedoch nicht im laufenden Projekt AI Börse verwendet. Mehr dazu siehe Praxisteil vom Herrn Mader.

⁹DesignPattern

¹⁰siehe 6.2.3

¹¹Floatwert wie sicher, das Ergebnis ist

Probleme mit Windows

Wenn man nun das vorher erklärte Python Skript auf dem Windowsserver des Projektes AI Börse ausführt, bekommt man eine Fehlermeldung.

Eine verkürzte Ausgabe der Fehlermeldung:

```

1 20/12/22 21:12:21 ERROR Executor: Exception in task 0.0 in stage 18.0 (TID
2   71)
3 java.io.IOException: Cannot run program "C:/Users/20160584/AppData/Local/
4   Programs/Python/Python36": CreateProcess error=5, Zugriff verweigert
5 at java.lang.ProcessBuilder.start(ProcessBuilder.java:1048)
6 at org.apache.spark.api.python.PythonWorkerFactory.createSimpleWorker(
7   PythonWorkerFactory.scala:155)
8 at org.apache.spark.api.python.PythonWorkerFactory.create(
9   PythonWorkerFactory.scala:97)
10 at org.apache.spark.SparkEnv.createPythonWorker(SparkEnv.scala:117)
11 at org.apache.spark.api.python.BasePythonRunner.compute(PythonRunner.scala
12   :109)
13 ... 28 more
14 at java.lang.Thread.run(Thread.java:748)
15 Caused by: java.io.IOException: CreateProcess error=5, Zugriff verweigert
16 at java.lang.ProcessImpl.create(Native Method)
17 at java.lang.ProcessImpl.<init>(ProcessImpl.java:444)
18 at java.lang.ProcessImpl.start(ProcessImpl.java:139)
19 at java.lang.ProcessBuilder.start(ProcessBuilder.java:1029)
20 [...]
21 ERFOLGREICH: Der Prozess mit PID 6088 (untergeordnetem Prozess von PID
22   2348) wurde beendet.
23 FEHLER: Der Prozess mit PID 2348 (untergeordnetem Prozess von PID 2396)
24   konnte nicht beendet werden.
25 Ursache: Dieser Vorgang wird nicht unterstützt.
26 FEHLER: Der Prozess mit PID 2396 (untergeordnetem Prozess von PID 1252)
27   konnte nicht beendet werden.
28 Ursache: Dieser Vorgang wird nicht unterstützt.

```

Listing 7.7: Output praxis Pipeline2

Aus der Fehlermeldung kann man herauslesen, dass ein Java Thread mit der ID 748 den Fehler `java.io.IOException: CreateProcess error=5, Zugriff verweigert`, das liegt daran, dass Java in einem Benutzerordner läuft, welcher keine Adminrechte hat und somit kann das Programm nicht auf alle Funktionen zugreifen. Auch nach Überarbeitung der Rollen und Recht aller betroffenen Ordnern, wurde die selbe Fehlermeldung ausgegeben.

Aus der Fehlermeldung kann man auch noch erkennen, dass am Ende nicht alle Hintergrundprozesse von Spark beendet werden konnten. Das ist der Grund, weswegen der `\temp12` Ordner mit vielen Dateien von bereits beendeten Prozessen überfüllt wird.

¹²Verzeichnis für verübergehende Dateien welche meistens nur für die Laufzeit eines bestimmten

Umstellung auf Linux

Daher einigte sich das Projektteam im Projekt AI Börse dafür, das Programm auf den Linuxserver auszulagern.

Hierfür muss zuerst alle notwendigen Pakete und Frameworks auf dem Server installiert werden. Das funktioniert im Gegensatz zu Windows sehr einfach man benötigt nur dieses kurze Skript:

```
1 sudo apt-get update -qq
2 sudo apt-get install -y openjdk-8-jdk-headless
3 java -version
4
5 python3 -m pip install pyspark==2.4.4
6 python3 -m pip install spark-nlp
```

Listing 7.8: Installation Spark auf Linux

Bei diesem Skript wird zuerst ein JDK mit der Version 1.8 installiert und anschließend über den Befehl `java -version` überprüft.

Danach wird PySpark und SparkNLP installiert, das kann man ganz einfach über `pip` repositiy machen.

Anschließend kann man das Programm am Linuxserver laufen lassen.

7.2.2 Alternative Einbindungen von Clustersystemen

Spark DataFrame

Die meisten Programme welche in dem Projekt AI Börse verwendet werden, benutzen Pandas DataFrame, zum Zwischenspeichern der Daten. Anstelle dieser Pandas DataFrame können auch die in Spark integrierten DataFrames verwendet werden.

Ein Beispiel für die Verarbeitung über Spark DataFrames. Könnte so aussehen:

```
1 from pyspark.sql import *
2 from pyspark.sql import functions as F
3
4 #Starten der Session
5 spark = SparkSession.builder.appName('BewertungBoersenbriefe').
6     getOrCreate()
7
8 #Erstellen von Testdaten
9 columns = ["link", "datum", "uhrzeit", "website_url", "sentence"]
10 data = [("https://www.onvista.de/news/dax-leichte-katerstimmung-442839755
11      ", "21.03.2021", "20:00", "www.onvista.de", "Nach dem gestrigen
12      Kurssprung an 14.800 Punkte herrscht am letzten Handelstag dieser
13      Woche Katerstimmung unter Anlegern, das Barometer tendiert merklich im
14      Minus."), ("https://www.onvista.de/news/dax-leichte-katerstimmung
15      -442839755", "21.03.2021", "20:01", "www.onvista.de", "Dieser verlor
16      im gestrigen Handel merklich an Wert, es geht die Angst steigender
17      Umlaufrenditen um.")]
18
19 #Aus den Testdaten einen Dataframe erstellen
20 sentences = spark.createDataFrame(data, columns)
21
22 #Ausgabe des Dataframes
23 print(sentences.show())
24
25 #Hinzufuegen einer neuen Reihe
26 newRow = [("https://www.onvista.de/news/dax-leichte-katerstimmung
27      -442839755", "21.03.2021", "20:20", "www.onvista.de", "Widerstände:
28      14.780 // 14.815 // 14.877 // 14.900 // 14.981 // 15.002")]
29 newSentence = spark.createDataFrame(newRow, columns)
30
31 sentences = sentences.union(newSentence)
32
33 print(sentences.show())
34
35 #Sortieren der Zeilen
36 print("Sortiert:")
37 print(sentences.sort(F.desc("sentence")).show())
38
39 #Iterieren
```

```

30 for row in sentences.collect():
31     print(row['sentence'])
32
33 #Hinzufuegen einer neuen Spalte
34 sentences = sentences.withColumn("sentiment", F.lit(None).cast('integer'))
35
36 print(sentences.show())

```

Listing 7.9: Benutzung PySpark DataFrames

Das Programm wurde über ein community Databrick Notebook geschrieben: (LR:Web26), 21.03.2021

Die Imports zu erstellen ist die erste Tätigkeit, welche erledigt werden muss. Dabei werden im Gegensatz zum obigen Programm über Pipelines nur ein Import für `pyspark.sql` benötigt, weil die DataFrames darüber laufen.

Danach wird eine neue Sparksession erstellt. Das geschieht über die Funktion `.builder`, darüber kann man der Session mehrere Attribute geben, in diesem Beispiel wird nur ein Name übergeben. Mit dem Zusatz `.getOrCreate()` wird zuerst nach einer offenen Session gesucht und wenn es keine gibt, wird erst eine neue erstellt.

Als nächster Schritt werden Testdaten definiert. Dabei wird ein Array aus den Spaltenüberschriften erstellt und ein zweidimensionales Array für die eigentlichen Datensätze.

Anschließend wird über den Befehl `.createDataFrame()` aus den Daten ein DataFrame erstellt mit den übergebenen Spaltenüberschriften.

Dieser DataFrame wird dann über den Befehl `.show()` ausgegeben. Das sieht so aus:

	link	datum uhrzeit	website_url	sentence
1	https://www.on...	21.03.2021 20:00	www.onvista.de Nach dem gestrige...	
2	https://www.on...	21.03.2021 20:01	www.onvista.de Dieser verlor im ...	

Listing 7.10: PySpark DataFrames Output1

Als nächstes wird eine neue Reihe an den DataFrame angehängt. Dafür muss man erstmal einen neuen DataFrame erstellen, welcher die selben Spalten hat. Danach kann man über den Befehl `.union()` zwei DataFrames vereinen. Dieser neu erstellte DataFrame sieht dann so aus:

	link	datum uhrzeit	website_url	sentence
1				
2				
3				

```

4 |https://www.on...|21.03.2021| 20:00|www.onvista.de|Nach dem gestrige...|
5 |https://www.on...|21.03.2021| 20:01|www.onvista.de|Dieser verlor im ...|
6 |https://www.on...|21.03.2021| 20:20|www.onvista.de|Widerstände: 14.7...|
7 +-----+-----+-----+-----+

```

Listing 7.11: PySpark DataFrames Output2

Da die DataFrames auch durch SQL Befehle bedient werden können, gibt es die Möglichkeit diese ganz einfach zu sortieren.

Die sortierte Ausgabe nach den Sätzen sieht dann so aus:

```

1 Sortiert:
2 +-----+-----+-----+-----+
3 |           link|      datum|uhrzeit| website_url|          sentence|
4 +-----+-----+-----+-----+
5 |https://www.on...|21.03.2021| 20:20|www.onvista.de|Widerstände: 14.7...|
6 |https://www.on...|21.03.2021| 20:00|www.onvista.de|Nach dem gestrige...|
7 |https://www.on...|21.03.2021| 20:01|www.onvista.de|Dieser verlor im ...|
8 +-----+-----+-----+-----+

```

Listing 7.12: PySpark DataFrames Output3

Es ist natürlich auch möglich mit einer For-Schleife über den DataFrame zu iterieren, dabei wird bei jeder Zeile der Satz ausgegeben.

Ausgabe dafür:

```

1 Nach dem gestrigen Kurssprung an 14.800 Punkte herrscht am letzten ...
2 Dieser verlor im gestrigen Handel merklich an Wert, es geht die Angst ...
3 Widerstände: 14.780 // 14.815 // 14.877 // 14.900 // 14.981 // 15.002

```

Listing 7.13: PySpark DataFrames Output4

Als letzter Schritt wird noch eine neue Spalte an den DataFrame angehängt. Diese Spalte wird mit Nullwerten gefüllt und dient dazu, im Laufe des Programms befüllt zu werden. Bis dahin sieht der neue DataFrame so aus:

```

1 +-----+-----+-----+-----+-----+-----+
2 |           link|      datum|uhrzeit| website_url|          sentence|sentiment|
3 +-----+-----+-----+-----+-----+-----+
4 |https://www.o...|21.03.2021| 20:00|www.onvista.de|Nach dem ...|      null|
5 |https://www.o...|21.03.2021| 20:01|www.onvista.de|Dieser ve...|      null|
6 |https://www.o...|21.03.2021| 20:20|www.onvista.de|Widerst  ...|      null|
7 +-----+-----+-----+-----+-----+-----+

```

Listing 7.14: PySpark DataFrames Output5

(LR:Web24, vgl. sparkbyexamples.com 21.03.2021)

(LR:Web25, vgl. towardsdatascience.com 21.03.2021)

Fazit

Wie das Programm beweist, ist es möglich über einen PySpark DataFrame die selben Arbeiten wie über einen Pandas DataFrame zu bewältigen.

Es ist auch noch möglich über PySpark einen Datenbankzugriff zu tätigen, somit braucht man dafür keine externe Library wie bei Pandas zu verwenden.

Ein solcher Datenbankzugriff sieht in vereinfachter Version so aus:

```
1 df = spark.read.jdbc(url=url,table='testdb.employee',properties=db_properties)
```

Listing 7.15: PySpark Datenbankzugriff

Damit der Code funktioniert benötigt man noch ein db_properties Datei:

```
1 url = jdbc:postgresql://10.128.7.12:5432/
2 Database = AI_Boerse
3 schema = public
4 username= postgres
5 password = postgres
6 driver=org.postgresql.Driver
```

Listing 7.16: db_properties

(LR:Article16, vgl. medium.com 21.03.2021)

Koalas

Es ist für die Programme, welche schon Pandas DataFrame API verwenden, auch möglich, Koalas zu implementieren. Somit müssen nur minimale Änderungen an dem bereits programmierten Code vorgenommen werden.

Um Koalas in einem Programm zu verwenden, müssen nur die folgenden Imports getätigigt werden:

```
1 import pandas as pd
2 import numpy as np
3 import databricks.koalas as ks
4 from pyspark.sql import SparkSession
```

Listing 7.17: Import Koalas

Danach ist es möglich, über den Stellvertreter ks, die Pandas API zum Verwalten eines Apache Spark DataFrame zu verwenden.

Ein Tutorial für die Einbindung von Koalas findet man auf der offiziellen Dokumentation:
(LR:Docs16, koalas.readthedocs.io 21.03.2021)

Kapitel 8

Computerlinguistik

8.1 Allgemeines

Computational linguistics is the scientific and engineering discipline concerned with understanding written and spoken language from a computational perspective, and building artifacts that usefully process and produce language, either in bulk or in a dialogue setting. (SM:Web01, Schubert 02.11.2020)

Der Mensch ist ein Wesen der Sprache. Dass er Sprachen erzeugen und verstehen kann, ist das was ihn ausmacht (SM:Web16, vgl. Friederici 19.01.2021). Die Computerlinguistik ist das Gebiet, das die Lehre der menschlichen Sprachen, die Linguistik, mit der Informatik verbindet. Sie ist aus unserem Alltag nicht mehr wegzudenken und ermöglicht viele praktische Anwendungen, wie das Abfragen von Terminen auf unserem Mobiltelefon per Spracherkennung oder das automatische Übersetzen von fremdsprachigen Webseiten. (SM:Web17, vgl. master-and-more.at 19.01.2021)

In den folgenden Kapiteln werden dem Leser die Grundsätze der Computerlinguistik nähergebracht und es wird versucht, dem Leser ein Verständnis für die Verarbeitung von natürlicher Sprache mithilfe von Computersystemen zu vermitteln.

8.1.1 Ziele der Computerlinguistik

Die Ziele der Computerlinguistik decken ein breites und vielfältiges Spektrum ab. Ein paar der wichtigsten Ziele sind die gezielte Suche nach einem bestimmten Thema in einem Text, die effiziente maschinelle Übersetzung, die Beantwortung von Fragen, das

Zusammenfassen von Texten oder die Analyse von Texten nach Thema, Gefühl oder anderen psychologischen Merkmalen. (SM:Web01, vgl. Schubert 02.11.2020)

Das ultimative Ziel ist die Kreation eines Computersystems, das sich auf dem Niveau eines Menschen ausdrücken, Sprachen lernen und Informationen aus Texten gewinnen kann. (SM:Web01, vgl. Schubert 02.11.2020)

8.2 Geschichte

8.2.1 1950 bis 1969

Die Computerlinguistik nahm ihre Anfänge in den USA in 1950er Jahren, als man sich zum Ziel setzte, fremdsprachige Texte, vor allem russische wissenschaftliche Zeitschriften, automatisch ins Englische zu übersetzen. (SM:Article01, vgl. Hutchins 01.01.2021)

Diese ersten Projekte, die sich mit der Übersetzung von russischen Texten befassten, wurden hauptsächlich vom US-amerikanischen Militär finanziert. Als in den 1950er und 1960er Jahren der Wettlauf ins All zwischen den Vereinigten Staaten und der Sowjetunion begann, konnten diese Projekte starke Förderungen erhalten. Nichtsdestotrotz waren die Ergebnisse der Arbeiten dieser Zeitperiode enttäuschend. Die Finanzierung solcher Projekte, die sich mit maschinellem Übersetzen beschäftigten, wurden 1966, mit der Veröffentlichung des ALPAC¹ Reports, großteils eingestellt. In diesem Report wurde dargelegt, dass die Vereinigten Staaten keinen Bedarf an maschinellen Übersetzungsprogrammen haben, solange sich die Qualität der Ergebnisse nicht um ein Vielfaches bessert, was damals als unwahrscheinlich angesehen wurde. (SM:Article01, vgl. Hutchins 01.01.2021)

Die linguistischen Probleme, auf die die Forscher gestoßen waren, hatten sich als größer herausgestellt als sie erwartet hatten und der Fortschritt war schmerhaft langsam. Man sollte darauf hinweisen, dass Yehoshua Bar-Hillel² im Jahr 1960 einen kritischen Bericht über die Forschung an maschineller Übersetzung veröffentlicht hatte, in dem er das Ziel der FAHQ³ ablehnte und die Entwicklung computerbasierter Systeme befürwortete, die für die Verwendung durch menschliche Übersetzer konzipiert waren. Er nannte dieses Zusammenspiel zwischen Übersetzer und Computer „Mensch-Maschine-Symbiose“. Die Verfasser des ALPAC Reports waren der gleichen Meinung wie Bar-Hillel und empfahlen, die Forschung an vollautomatischen Systemen einzustellen und die Aufmerksamkeit auf

¹Automatic Language Processing Advisory Committee

²Israelischer Linguist, bekannt für seine grundlegenden Arbeiten zur maschinellen Übersetzung

³Fully automatic high quality translation (ger.: vollautomatische Übersetzung in hoher Qualität)

Hilfsmittel für menschliche Übersetzer auf niedrigerem Niveau zu richten. (SM:Article01, vgl. Hutchins 01.01.2021)

8.2.2 1970 bis 1989

In den Jahren nach der Veröffentlichung des ALPAC Reports, wurde die Forschung an maschineller Übersetzung nicht komplett eingestellt, sondern lief in stark reduziertem Umfang weiter. In den 1970er Jahren konnten einige Erfolge verzeichnet werden. Im Jahr 1970 begann die US-Luftwaffe das SYSTRAN-System für russisch-englische Übersetzungen zu nutzen. In Kanada begann man im Jahr 1976 das METEO-System zu nutzen, um französische Wettervorhersagen automatisch ins Englische und englische Wettervorhersagen ins Französische zu übersetzen. (SM:Article01, vgl. Hutchins 01.01.2021)

Am wichtigsten für die damalige Forschung war vermutlich der Erwerb der englisch-französischen Version von SYSTRAN durch die Europäische Kommission im Jahr 1976. Das System wurde daraufhin weiterentwickelt, um weitere europäische Sprachen zu unterstützen, wobei Englisch, Französisch, Deutsch und Spanisch die Hauptrollen einnahmen. (SM:Article02, vgl. Petrits 01.01.2021)

In den 1980er Jahren, konnte die Forschung an maschineller Übersetzung wieder einen Aufschwung erfahren. Überall auf der Welt - vor allem in Japan - begann man wieder an neuen Ideen zu forschen. Neue Quellen finanzieller Unterstützung ergaben sich durch die Europäische Union und große Technikunternehmen. (SM:Article01, vgl. Hutchins 01.01.2021)

8.2.3 Ab 1990

Ein Umschwung in der Forschung der Computerlinguistik konnte in den 1990er Jahren beobachtet werden, als man anfing korpusbasierte Methoden zu verwenden (SM:Article01, vgl. Hutchins 16.01.2021). Korpusbasierte Methoden basieren auf der mathematischen Analyse von linguistischen Phänomenen, die aus der Analyse von riesigen Mengen an Text, sogenannten Textkorpora, gewonnen werden (SM:Web05, vgl. Expert System Team 16.01.2021).

Vor allem die, durch das *IBM Candide Project* eingeführten, statistischen Ansätze (siehe Kapitel 9.5) konnten vielversprechende Ergebnisse liefern und die Grenzen der bisher verwendeten regelbasierten Methoden (symbolische Ansätze, siehe Kapitel 9.4) aufzeigen. Probleme in den Bereichen der kontextbezogenen Semantik⁴ und der

⁴Bedeutungslehre

Anaphorik (siehe Kapitel 9.2.5) waren mithilfe von korpusbasierten Methoden leichter nachzuvollziehen. (SM:Article01, vgl. Hutchins 16.01.2021)

Im Jahr 2006 veröffentlichte Google seinen eigenen Übersetzungsdiest *Google Translate*, der auf statistischen Übersetzungsmethoden aufbaute. Er übersetzt Texte von einer in die andere Sprache, indem er sie zuerst ins Englische übersetzt, um sie dann wiederum in die Zielsprache zu übersetzen. (SM:Web06, vgl. Sommerlad 17.01.2021)

Im Jahr 2017 wurde der *Google Translate*-Dienst von statistischen Übersetzungsmethoden auf Methoden, die ein künstliches neuronales Netz nutzen (siehe Kapitel 9.6), umgestellt. Google setzt dabei auf rekurrente neuronale Netze (SM:Web08, vgl. heise.de 17.01.2021). Die Genauigkeit der Übersetzungen des Dienstes konnte damit um 60 % gesteigert werden. (SM:Web07, vgl. McGuire 17.01.2021)

Im selben Jahr veröffentlichte die deutsche DeepL GmbH ihren Übersetzungsdiest *DeepL*. Dieser Dienst basiert auf Convolutional Neural Networks und kann dadurch um ein Vielfaches bessere Übersetzungen (SM:Web09, vgl. deepl.com 17.01.2021) liefern als andere Übersetzungsdiene, wie *Google Translate*. (SM:Web08, vgl. heise.de 17.01.2021)

Heutzutage findet man Computerlinguistik in vielen Bereichen des Alltags, von der automatischen Spracherkennung in Sprachassistenten, wie Google Assistant oder Amazon Alexa, bis hin zur automatischen Übersetzung von Webseiten.

Kapitel 9

Natural Language Processing

Unter Natural Language Processing¹, einem Teilgebiet der Computerlinguistik, versteht man die Verbindung von künstlicher Intelligenz mit der Linguistik. (SM:Article04, vgl. Journal of the American Medical Informatics Association 02.01.2021)

Da, laut IBM, 2,5 Exabytes² an Daten alleine im Jahr 2017 generiert wurden und es sich bei etwa 80 % dieser Daten (SM:Video01, vgl. edureka! 06.01.2021) um unstrukturierte Daten (zum Beispiel Freitextdaten) handelt, nimmt Natural Language Processing eine immer wichtigere Rolle ein. (SM:Article06, vgl. Ganegedara, 06.01.2021)

9.1 Allgemeines

Natural Language Processing is a theoretically motivated range of computational techniques for analyzing and representing naturally occurring texts at one or more levels of linguistic analysis for the purpose of achieving human-like language processing for a range of tasks or applications. (SM:Article03, Liddy 02.01.2021)

Da es sich bei Natural Language Processing (NLP) um ein ständig wandelndes Gebiet handelt, gibt es keine einheitliche Definition, der alle zustimmen würden. Bei der oben zitierten Definition handelt es sich um eine von vielen, wobei diese die wichtigsten Aspekte beinhaltet:

¹Zu Deutsch etwa „maschinelle Verarbeitung natürlicher Sprache“

²1 Exabyte = 1,000,000,000 Gigabytes

- “*range of computational techniques*”:

Diese vage Ausdrucksweise ist notwendig, da es sehr viele verschiedene Ansätze und Methoden gibt, eine Sprache zu analysieren bzw. zu verarbeiten.

- “*Naturally occurring texts*”:

Diese Texte können in jeder Form von Sprache vorliegen, egal ob geschrieben oder gesprochen. Die einzige Voraussetzung ist, dass sie in einer Sprache geschrieben sind, die von den Menschen benutzt wird, um miteinander zu kommunizieren.

- “*levels of linguistic analysis*”:

Unter diesen „Ebenen der sprachlichen Analyse“ versteht man die verschiedenen Arten der Sprachverarbeitung, die am Werk sind, wenn Menschen miteinander kommunizieren. Während ein Mensch beim Kommunizieren alle diese Ebenen gleichzeitig verwendet, können NLP Systeme dies nicht, und sind deswegen meist auf einige wenige Ebenen bzw. Kombinationen von Ebenen spezialisiert. (Auf diese Ebenen wird näher in Kapitel 9.2 eingegangen)

- “*Human-like language processing*”:

Diese Aussage ist wichtig, da klar sein soll, dass NLP ein Gebiet der künstlichen Intelligenz ist.

(SM:Article03, vgl. Liddy 02.01.2021)

9.2 Ebenen von Natural Language Processing

Die Analyse eines Textes in NLP kann traditionell in einzelne Ebenen unterteilt werden. Die wichtigsten Rollen spielen dabei die Ebenen der Syntax, Semantik und Pragmatik. Der Kerngedanke bei dieser Herangehensweise ist, zuerst die Struktur (Syntax) des Textes zu analysieren, damit man mithilfe dieser Struktur eine semantische Analyse durchführen kann, die die Bedeutung des Satzes deuten soll. Zu guter Letzt wird eine pragmatische Analyse durchgeführt, die sich zum Ziel setzt, satzübergreifende (Diskurs) oder kontextbezogene Bedeutungen zu ermitteln. (SM:Article07, vgl. Indurkhyia und Damerau 17.01.2021)

9.2.1 Syntax

Die Syntax beschäftigt sich mit der Analyse von Wörtern in einem Satz, sodass die grammatischen Strukturen des Satzes aufgedeckt werden. (SM:Article03, vgl. Liddy 04.01.2021)

Die Ausgabe dieser Ebene ist eine, möglicherweise nicht lineare, Darstellung der Beziehungen zwischen den Wörtern eines Satzes. Die Syntax eines Satzes ist ausschlaggebend für seine Aussage. Die englischen Sätze „The dog chased the cat.“ und „The cat chased the dog.“ zum Beispiel, sind aus den exakt gleichen Wörtern aufgebaut, unterscheiden sich allerdings in der Syntax und sagen dadurch etwas komplett anderes aus. (SM:Article03, vgl. Liddy 04.01.2021)

Die Analyse des Satzes „The dog chased the cat.“ sieht folgendermaßen aus:

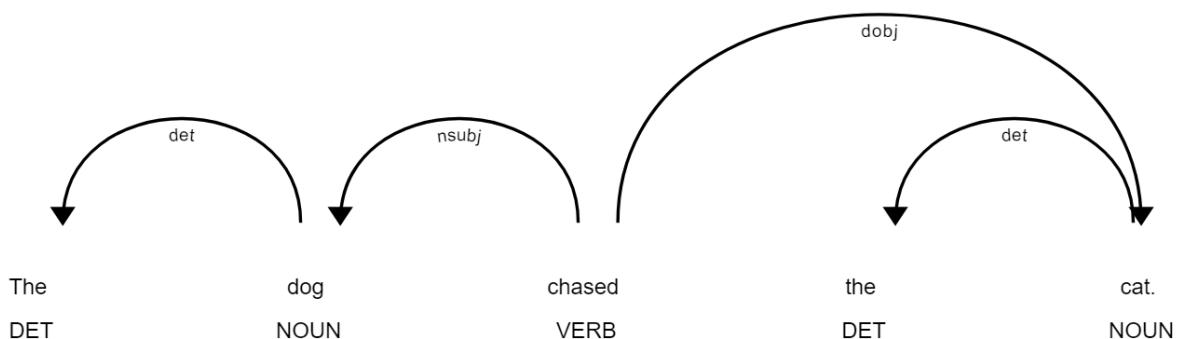


Abbildung 9.1: Syntax von „The dog chased the cat.“

Diese Grafik ist die Ausgabe folgenden Codes:

```

1 import spacy
2 from spacy import displacy
3
4 nlp = spacy.load("en_core_web_sm")
5 sentence = "The dog chased the cat."
6 doc = nlp(sentence)
7 displacy.render(doc, style = "dep")

```

Listing 9.1: Syntaxanalyse mithilfe von spaCy

Dieser Syntaxanalyse kann man nun entnehmen, dass das Wort „dog“ die Beziehung *nsubj* (= *nominal subject*) zum Wort „chased“ hat, und „cat“ in der Beziehung *dobj* (= *direct object*) zum Wort „chased“ steht. Also weiß man nun, dass „dog“ das Subjekt, „cat“ das Objekt und „chased“ das Verb, das die beiden verbindet, ist. Das heißt, der Hund hat die Katze gejagt und nicht andersherum. (SM:Web03, vgl. spaCy Dokumentation 04.01.2021)

9.2.2 Morphologie

In der Morphologie wird die Zusammensetzung von Wörtern analysiert, also aus welchen Worteinheiten ein Wort aufgebaut wird. Die kleinste Einheit, der man eine Bedeutung zuordnen kann, nennt man Morphem.

Das Wort „Rhabarberkuchenbäckerin“ zum Beispiel, besteht aus den Morphemen „Rhabarber“, „kuchen“, „bäcker“ und „in“.

Morpheme kann man in folgende Gruppen einteilen:

- *Wurzel*: Wurzel-Elemente sind Morpheme die eigenständig vorkommen können, wie z. B. „Bäcker“.
- *Affix*: Affixe sind Morpheme die nicht selbstständig vorkommen können. Bei dem Wort „kindlich“ wäre z. B. „lich“ ein Affix, genauer gesagt ein Suffix. Bei Affixen kann man noch unterscheiden zwischen:
 - *Präfix*, wenn es vor einer Wurzel steht.
 - *Suffix*, wenn es nach einer Wurzel steht.

(SM:Web02, vgl. Becker 03.01.2021)

9.2.3 Semantik

Mithilfe einer semantischen Analyse werden alle möglichen Bedeutungen eines Satzes bestimmt, indem man sich die Bedeutungen einzelner Wörter genauer ansieht und versucht einen Zusammenhang zwischen ihnen zu finden.

Das Problem bei allen Sprachen ist, dass es Wörter gibt, denen man mehrere Bedeutungen zuordnen kann. Durch die Ebene der Semantik wird versucht, mehrdeutigen Wörtern, ihre eindeutige Bedeutung innerhalb eines Satzes zuzuordnen.

(SM:Article03, vgl. Liddy 04.01.2021)

Man kann dabei zwischen lexikalischer und kompositioneller Semantik unterscheiden:

- Die *lexikalische Semantik* gleicht einzelne Wörter eines Textes mit einem Lexikon ab, um alle möglichen Bedeutungen der Wörter zu erfahren. In einem Lexikon ist der Wortschatz einer Sprache, im Hinblick auf ihre interne Bedeutungsstruktur, gespeichert.

- Die *kompositionelle Semantik* versucht die Bedeutung eines Wortes, im Hinblick auf die Syntax des Satzes, in dem es vorkommt, zu bestimmen.

(SM:Article08, vgl. Hausser 17.01.2021)

Um eine semantische Analyse durchführen zu können, muss zuvor die Syntax des Textes analysiert werden, damit aus den syntaktisch analysierten Ausdrücken eine semantische Repräsentation erstellt werden kann. Diese Repräsentation folgt der Funktor-Argument-Struktur. (SM:Article08, vgl. Hausser 18.01.2021)

Funktor-Argument-Struktur

In der Semantik wird davon ausgegangen, dass man die Bedeutung einfach strukturierte, sowie komplexer Sätze, auf dieselbe Art und Weise darstellen kann. Dabei folgt man der Funktor-Argument-Struktur. (SM:Web10, vgl. Breindl 18.01.2021)

Die Idee hinter der Funktor-Argument-Struktur ist, die Eigenschaft eines Wortes als Funktor herzunehmen und das Wort als Argument des Funktors zu verwenden.
(SM:Article09, vgl. Hackmack 18.01.2021)

Der Satz „Philip ist männlich.“ sieht in der Funktor-Argument-Struktur folgendermaßen aus:



Abbildung 9.2: Funktor-Argument-Struktur von „Philip ist männlich“
Quelle: (SM:Article09, vgl. Hackmack 18.01.2021)

Einem Funktor können auch mehr als ein Argument übergeben werden. Der Text „Fido ist ein Hund. Fido frisst Fleisch.“ kann wie folgt dargestellt werden:

hund(fido)
frisst(fido, fleisch)

(SM:Article10, vgl. Clematide 18.01.2021)

9.2.4 Pragmatik

Die Pragmatiklinguistik beschäftigt sich mit der Bestimmung der Bedeutung von Wörtern in einem Satz, deren Bedeutung vom Kontext abhängig ist.

In dem Satz

*Die Stadträte genehmigten die Demonstration der Demonstranten nicht, da **sie** Angst vor Gewalt hatten.*

weiß man ohne Kontext nicht, ob sich das Wort *sie* auf die Stadträte oder die Demonstranten bezieht. (SM:Article03, vgl. Liddy 04.01.2021)

Ein weiteres Gebiet der Pragmatiklinguistik ist die Bestimmung der Bedeutung von ironischen Äußerungen.

Der Satz

Das hast Du ja mal wieder ganz toll hingekriegt!

könnte einerseits ein Lob, andererseits, je nach Betonung, auch eine negative Aussage sein. (SM:Article05, vgl. Lapp 04.01.2021)

Um solch eine Analyse durchführen zu können, braucht das NLP-System viel Wissen über die reale Welt, damit es den Kontext ermitteln kann. (SM:Article03, vgl. Liddy 04.01.2021)

9.2.5 Diskurs

Die Diskursanalyse beschäftigt sich nicht mit einzelnen Sätzen oder Wörtern, sondern mit einem Text aus mehreren Sätzen. Sie versucht satzübergreifende Bedeutungen zu bestimmen. (SM:Article03, vgl. Liddy 04.01.2021)

Eine Herangehensweise bei der Diskursanalyse, ist mithilfe der **Anaphorik** möglich. Dabei werden Satzteile, die, wenn sie alleine stehen, keine Bedeutung haben, mit den Satzteilen aus einem anderen Satz ersetzt, auf die sie „anaphorisch“ verweisen. (SM:Article03, vgl. Liddy 04.01.2021)

Bei dem Text

Peter geht heute spazieren. **Er** geht gerne durch den Wald.

verweist das Pronomen *Er* im zweiten Satz anaphorisch auf den Namen *Peter* im ersten Satz. Die Anaphorik macht also aus dem Text Folgendes:

Peter geht heute spazieren. **Peter** geht gerne durch den Wald.

Diskurs-/Textstrukturerkennung

Eine weitere Variante der Diskursanalyse ist die Diskurs-/Textstrukturerkennung. Ihre Aufgabe ist die Funktion einzelner Sätze innerhalb eines Satzes zu bestimmen. Zeitungsartikel können dadurch zum Beispiel in die Diskurskomponenten *Einleitung*, *Hauptteil*, *Vorgeschichte*, *Bewertung*, *Zitate* und *Erwartung* zerlegt werden. (SM:Article03, vgl. Liddy 04.01.2021)

9.2.6 Phonologie

Die Phonologie beschäftigt sich mit der Interpretation von Sprachklängen innerhalb oder zwischen Wörtern. Dabei werden drei Bereiche unterschieden:

- *Phonetik* beschäftigt sich mit Klängen innerhalb eines Wortes.
- *Phonemik* beschäftigt sich mit Klängen, wenn verschiedene Wörter kombiniert werden.
- *Prosodie* beschäftigt sich mit den verschiedenen Betonungen, die während der Aussprache eines Satzes auftreten.

(SM:Article03, vgl. Liddy 03.01.2021)

9.3 Aufgaben von Natural Language Processing

9.3.1 Tokenisierung

Bei der Tokenisierung geht es darum, einen Text in kleinere Einheiten (Token), zum Beispiel einzelne Wörter oder Sätze, aufzuteilen. Auch wenn diese Aufgabe trivial wirkt, kann sie eine Herausforderung darstellen. Bei Sprachen wie Japanisch, in denen Wörter nicht durch Leerzeichen getrennt werden, kann es zum Beispiel schwierig sein einzelne Wörter zu identifizieren. (SM:Article06, vgl. Ganegedara 06.01.2021)

Folgendes Beispiel demonstriert die Tokenisierung mithilfe der Python-Library spaCy:

```
1 import spacy
2
3 nlp = spacy.load("de_core_news_sm")
4 doc = nlp("1 kg Äpfel kostet im Supermarkt 1,99 €.")
5 tokens = []
6 for token in doc:
7     tokens.append(token.text)
8 print(tokens)
```

Listing 9.2: Tokenisierung mithilfe von spaCy

Die Ausgabe dieses Codes sieht folgendermaßen aus:

```
['1', 'kg', 'Äpfel', 'kostet', 'im', 'Supermarkt', '1,99', '€', '.']
```

9.3.2 Named Entity Recognition

Mithilfe von Named Entity Recognition (NER) wird versucht Wörter in Kategorien, wie *Name*, *Zahl*, *Organisation*, *Zeit*, usw., einzuteilen. (SM:Article06, vgl. Ganegedara 06.01.2021)

Der Satz

John gab Mary am Montag in der Schule zwei Äpfel.

könnte mithilfe von NER, folgendermaßen transformiert werden:

[John]_{Name} gab [Mary]_{Name} am [Montag]_{Zeit} in der [Schule]_{Organisation} [zwei]_{Zahl} Äpfel.

(SM:Article06, vgl. Ganegedara 06.01.2021)

Damit man mit selbstdefinierten Kategorien eine Named Entity Recognition durchführen kann, muss man zuvor einen sogenannten *Entity Extractor* trainieren (SM:Web13, vgl. Roldós 19.01.2021). Wenn man, mithilfe der Python-Library spaCy, einen *Entity Extractor* trainieren will, muss man diesen mit Trainingsdaten füttern. Die Trainingsdaten werden in Form einer Liste aus Tupeln bereitgestellt. So ein Tupel könnte folgendermaßen aussehen:

```
("Walmart is a leading e-commerce company", {"entities": [(0, 7, "ORG")]} )
```

Durch diesen Tupel lernt der *Entity Extractor*, dass in dem Satz „Walmart is a leading e-commerce company“ von Index 0 (inklusive) bis Index 7 (exklusive) eine Organisation (ORG) zu finden ist. Damit der *Entity Extractor* bedeutungsvolle Erkenntnisse aus den Trainingsdaten gewinnen kann, sollte die Liste der Tupel mehrere Hundert Einträge beinhalten.

(SM:Web15, vgl. Shrivarsheni 19.01.2021)

9.3.3 Part-of-Speech-Tagging

Beim Part-of-Speech-Tagging (POS-Tagging) wird jedem Wort in einem Satz eine Wortart/ein POS-Tag zugeordnet. So ein POS-Tag könnte zum Beispiel ADJA heißen und alle Adjektive kennzeichnen. (SM:Web04, vgl. Reichel 06.01.2021)

Bei diesem Schritt, werden Wörtern, zu denen mehrere POS-Tags passen könnten, der wahrscheinlichste POS-Tag, aufgrund des Kontexts, zugeordnet. (SM:Article03, vgl. Liddy 06.01.2021)

Mithilfe der Python-Library spaCy kann man sich die POS-Tags der einzelnen Wörter eines Satzes folgendermaßen anzeigen lassen:

```
1 import spacy
2
3 nlp = spacy.load("de_core_news_sm")
4 doc = nlp("1 kg Äpfel kostet im Supermarkt 1,99 €.")
5 tokens = []
6 for token in doc:
7     tokens.append([token.text, token.pos_])
8 print(tokens)
```

Listing 9.3: Part-of-Speech-Tagging mithilfe von spaCy

Folgende Ausgabe wird durch diesen Code generiert:

```
[['1', 'NUM'],
['kg', 'X'],
['Äpfel', 'NOUN'],
['kostet', 'VERB'],
['im', 'ADP'],
['Supermarkt', 'NOUN'],
['1,99', 'NUM'],
['€', 'ADJ'],
['.', 'PUNCT']]
```

Wie man hier sehen kann, ist das POS-Tagging des deutschen spacy Modells `de_core_news_sm` nicht perfekt. Es erkennt zum Beispiel nicht, dass es sich bei dem Token `kg` um die Abkürzung für Kilogramm, also einem Nomen, handelt, und weist im, statt dem POS-Tag `NOUN`, den POS-Tag `X` zu (`X` steht für `other` und wird den Wörtern zugewiesen, die nicht erkannt werden). Zudem wird dem Eurozeichen der POS-Tag `ADJ` (für Adjektiv) zugewiesen, obwohl es sich dabei nicht um ein Adjektiv handelt.

9.3.4 Text-Klassifikation

Die Text-Klassifikation hat viele Aufgaben, wie zum Beispiel die Gefühlsanalyse, die Spamerkennung, die Einteilung von Zeitungsartikeln in Kategorien wie Sport, Politik, usw., das Erkennen ob Produktbewertungen negativ oder positiv sind, und viele mehr. (SM:Article06, vgl. Ganegedara 06.01.2021)

Text-Kategorisierung

Damit ein NLP-System zum Beispiel Zeitungsartikel in Kategorien, wie Sport und Politik, einteilen kann, muss zuvor ein *Text Classification Model* trainiert werden. (SM:Web11, vgl. monkeylearn.com 18.01.2021)

Wird der symbolische Ansatz verwendet (siehe Kapitel 9.4), also ein regelbasiertes System aufgebaut, müssen zuvor zwei Listen definiert werden. Diese Listen enthalten jeweils häufig vorkommende Stichwörter der jeweiligen Kategorie. So könnte die Liste für die Kategorie Sport die Wörter *Fußball*, *Basketball* und *LeBron James* enthalten, während die Liste für die Kategorie Politik die Wörter *Donald Trump*, *Hillary Clinton* und *Putin* enthält. Das *Text Classification Model* wird daraufhin mit diesen Listen trainiert. (SM:Web11, vgl. monkeylearn.com 18.01.2021)

Wenn man nun den Text „Wann ist LeBron James' erstes Spiel mit den Lakers?“ mithilfe dieses trainierten Modells analysiert, wird dieser in die Kategorie Sport eingeteilt,

da mehr Stichwörter aus der sportbezogenen Liste im Text vorkommen als aus der politikbezogenen Liste. (SM:Web11, vgl. monkeylearn.com 18.01.2021)

9.3.5 Textgenerierung

Bei der Textgenerierung, auch Natural Language Generation (NLG) (SM:Web14, vgl. Joshi 19.01.2021) genannt, wird ein Modell (zum Beispiel ein neuronales Netz) mit einer riesigen Sammlung an Textdokumenten trainiert (SM:Article06, vgl. Ganegedara 06.01.2021). Dieses Modell wird als *Language Model* bezeichnet und ist nach dem Training dazu in der Lage, die Wahrscheinlichkeit, mit der eine gewisse Kombination von Wörtern in der natürlichen Sprache vorkommt, zu berechnen (SM:Web14, vgl. Joshi 19.01.2021).

Somit kann das *Language Model* sagen, dass der Satz „Die Katze ist klein.“ zu einer höheren Wahrscheinlichkeit in einem Text vorkommt als der Satz „Klein ist die Katze.“. Bei der Textgenerierung sollte aber auch eine Diskursanalyse durchgeführt werden, damit man zum Beispiel satzübergreifende Vorhersagen für das nächste Wort treffen kann (SM:Web14, vgl. Joshi 19.01.2021).

“Alice went to the beach. There she built a sandcastle”



Abbildung 9.3: Diskursanalyse bei der Textgenerierung
Quelle: (SM:Web14, vgl. Joshi 19.01.2021)

Wenn man diesen Satz als Beispiel hennimmt, sollte das *Language Model* eines Auto vervollständigungs-Systems dazu in der Lage sein, das Wort „sandcastle“, aufgrund des Wortes „beach“ im vorhergehenden Satz, vorherzusagen. (SM:Web14, vgl. Joshi 19.01.2021)

Mithilfe der Textgenerierung kann man also Auto vervollständigung umsetzen. Sie ist allerdings auch ein wesentlicher Bestandteil von Chatbots und persönlichen Assistenten (SM:Web14, vgl. Joshi 19.01.2021). Eine weitere Anwendungsmöglichkeit wäre die Generierung einer Science-Fiction-Geschichte, wenn das Modell zuvor mit bereits vorhandenen Science-Fiction-Geschichten trainiert wurde. (SM:Article06, vgl. Ganegedara 06.01.2021)

9.3.6 Question Answering

Beim Question Answering (QA) geht es, wie der Name schon sagt, darum Antworten zu individuellen Fragen zu generieren. Es ist die Grundlage für moderne Sprachassistenten und Chatbots und hat einen hohen kommerziellen Wert. (SM:Article06, vgl. Ganegedara 16.01.2021)

Wenn man einen persönlichen Assistenten, wie Google Assistant, die Frage

Wie wird das Wetter morgen?

stellt, kann dieser mithilfe von Natural Language Processing die Bedeutung der Frage erkennen. Die semantische Analyse kann aus dem syntaktischen Aufbau der Frage, die Funktor-Argument-Struktur

thema (wetter) und datum (morgen)

ermitteln. Aus dieser Struktur kann der persönliche Assistent nun erkennen, dass es um das Wetter morgen geht. Durch den Einsatz von Textgenerierung kann er zu guter Letzt dem Benutzer eine Antwort anzeigen.

(SM:Web12, vgl. Wolff 18.01.2021)



Abbildung 9.4: Question Answering des Google Assistant

9.3.7 Maschinelle Übersetzung

Das Gebiet der maschinellen Übersetzung ist eines der ersten Gebiete mit dem sich die Computerlinguistik bzw. NLP beschäftigte. In diesem Gebiet geht es darum, einen Satz oder einen Text von einer Ausgangssprache in eine Zielsprache zu übersetzen. Was sich bei diesem Gebiet als die größte Herausforderung herausstellt, sind die Unterschiede im Sprachaufbau verschiedener Sprachen, welche eine Eins-zu-eins-Übersetzung, von einer Sprache in eine andere, nicht zulassen. (SM:Article06, vgl. Ganegedara 16.01.2021)

Die maschinelle Übersetzung des Satzes „Die Vögel zwitschern in der Früh“, mithilfe der Python-Library textblob, sieht folgendermaßen aus:

```
1 from textblob import TextBlob  
2 blob = TextBlob("Die Vögel zwitschern in der Früh.")  
3 print(blob.translate(to = "en"))
```

Listing 9.4: Maschinelle Übersetzung mithilfe von textblob

Ausgabe:

The birds chirp in the morning.

9.4 Symbolisches Natural Language Processing

Bei der Entwicklung eines NLP-Systems können verschiedene Ansätze zur Lösung verschiedener Probleme herangezogen werden. Grundsätzlich kann man dabei zwischen dem *symbolischen* und dem *statistischen* Lösungsansatz unterscheiden.

Auf die Frage, welcher der beiden Ansätze der bessere ist, gibt es keine eindeutige Antwort. Je nachdem was für ein Problem es zu lösen gibt, sollte entschieden werden, an welchem Ansatz man sein System orientiert. (SM:Web19, vgl. Couto 27.01.2021)

In den folgenden Kapiteln wird auf die verschiedenen Ansätze näher eingegangen.

9.4.1 Allgemeines

The symbolic approach to natural language processing is based on human-developed rules and lexicons. In other words, the basis behind this approach is in generally accepted rules of speech within a given language which are materialized and recorded by linguistic experts for computer systems to follow. (SM:Web05, Expert System Team 27.01.2021)

Beim symbolischen Ansatz handelt es sich um den ältesten Lösungsansatz im Gebiet des Natural Language Processings. Er basiert auf von Menschen bzw. Linguisten aufgestellten Regeln, die Regelmäßigkeiten bzw. Muster in einer natürlichen Sprache beschreiben sollen. Daher spricht man beim symbolischen Lösungsansatz auch vom *regelbasierten* Ansatz. (SM:Web18, vgl. Mayo 27.01.2021)

Tendenziell eignen sich symbolische Systeme besser für Probleme, bei denen die Eingangsdaten einen immer ähnlichen Aufbau haben. Dadurch kann ein Mensch Muster erkennen und Regeln definieren, nach denen das System handeln soll. (SM:Web19, vgl. Couto 27.01.2021)

Der größte Vorteil von symbolischen NLP-Systemen im Vergleich zu statistischen, ist die Transparenz. Der Computerlinguist, der die Regeln für das symbolische NLP-System aufgestellt hat, weiß genau warum sein System so reagiert wie es reagiert. Bei einem statistischen System, das durch Methoden des Machine Learnings trainiert wurde, ist dies nicht immer klar. (SM:Web57, vgl. inbenta.com 19.03.2021)

9.5 Statistisches Natural Language Processing

9.5.1 Allgemeines

The statistical approach to natural language processing is based on observable and recurring examples of linguistic phenomena. Models based on statistics recognize recurring themes through mathematical analysis of large text corpora. By identifying trends in large samples of text the computer system can develop its own linguistic rules that it will use to analyze future input and/or the generation of language output. (SM:Web05, Expert System Team 27.01.2021)

Im Gegensatz zum symbolischen Ansatz basiert der statistische Lösungsansatz nicht auf von Menschen aufgestellten Regeln, sondern auf, mithilfe von mathematischen Verfahren, berechneten Regeln. Dabei werden meist riesige Textkorpora herangezogen, die dem System als Eingabedaten bereitgestellt werden. In diesen Unmengen an Texten kann das System dann Muster und linguistische Phänomene erkennen und diese nutzen um ein statistisches Modell zu trainieren. Aus diesem Grund nennt man den statistischen Ansatz auch *korpusbasierten* Ansatz. (SM:Article03, vgl. Liddy 28.01.2021)

Statistische Modelle können für einfache Anwendung noch händisch aufgestellt werden, für komplexere Anwendung müssen diese jedoch mithilfe von *Machine Learning* Algorithmen aufgestellt werden (siehe Kapitel 9.5.2).

Den praktischen Unterschied zwischen dem symbolischen und dem statistischen Ansatz kann man gut anhand der Aufgabe des Part-of-Speech-Taggings (POS-Taggings) veranschaulichen:

Angenommen man will die wahrscheinlichsten POS-Tags für die Wörter des Satzes

John bought a book.

finden, kann man dies mit beiden Ansätzen umsetzen.

In einem symbolischen NLP-System könnte das POS-Tagging mithilfe eines BRILL-Taggers durchgeführt werden. Der BRILL-Tagger ist vermutlich der bekannteste regelbasierte POS-Tagger. Er weist dem ersten Wort in einem Satz seinen wahrscheinlichsten POS-Tag zu, ohne auf den Kontext zu achten. Jedem weiteren Wort wird dann ein POS-Tag, mit Rücksicht auf den vorhergehenden POS-Tag, zugewiesen. Dazu braucht der POS-Tag eine, von einem Menschen definierte, Liste an Regeln.

So eine Regel könnte folgendermaßen aussehen:

Adj → Verb falls der vorhergehende Tag ProperNoun war

Diese Regel wird vom BRILL-Tagger folgendermaßen interpretiert: Wenn der wahrscheinlichste POS-Tag eines Wortes *Adj* wäre, aber der vorhergehende Tag *ProperNoun* war, weise dem Wort stattdessen den Tag *Verb* zu.

Falls in diesem Beispielsatz dem Wort „*bought*“ der Tag *Adj* zugewiesen wird, kann dies durch diese vordefinierte Regel korrigiert werden, da der wahrscheinlichste POS-Tag für das vorhergehende Wort „*John*“, *ProperNoun* ist.

(SM:Web19, vgl. Couto 31.01.2021)

In einem statistischen NLP-System wird das POS-Tagging als ein *Sequence Labeling* Problem angesehen. Statt sich nur den POS-Tag des vorhergehenden Wortes anzusehen, betrachtet ein statistisches System die POS-Tags aller bisherigen Wörter des Satzes. Für unseren Beispielsatz könnte so ein System, wenn es bereits die POS-Tags für „*John bought a*“ bestimmt hat, für das Wort „*book*“ mit Sicherheit sagen, dass es sich dabei um ein Nomen und nicht um ein Verb handelt. (SM:Web19, vgl. Couto 31.01.2021)

9.5.2 Machine Learning

Machine learning focuses on applications that learn from experience and improve their decision-making or predictive accuracy over time. (SM:Web23, ibm.com 01.02.2021)

Beim Machine Learning³ handelt es sich um ein Teilgebiet der künstlichen Intelligenz, in dem es darum geht, einem Programm/einem Algorithmus beizubringen, Muster in riesigen Mengen an Daten zu erkennen. Nachdem das Programm mit gegebenen Daten trainiert wurde, kann es, aufgrund der erkannten Muster, Entscheidungen für neue Daten treffen. (SM:Web23, vgl. ibm.com 01.02.2021)

Machine Learning Prozess

Den Prozess des Machine Learnings kann man grob in 4 Schritte einteilen:

1. Auswahl und Aufbereitung der Trainingsdaten

Als Allererstes werden aus riesigen Mengen an Daten, die Daten ausgewählt mit denen der Algorithmus trainiert werden soll. Bei diesen Daten handelt es sich, im Fall von *Supervised Learning*, um *labeled data* oder, bei im Fall von *Unsupervised Learning*, um *unlabeled data*.

Bei *labeled data* ist die vorherzusagende Größe bereits in den Trainingsdaten enthalten. Somit weiß der Algorithmus für jeden Datensatz wie die vorherzusagende Größe auszusehen hat, wenn man die anderen Werte des Datensatzes gegeben hat. Bei *unlabeled data* hingegen, ist keine vorherzusagende Größe in den Trainingsdaten definiert und der Algorithmus muss selber herausfinden, welche Eigenschaften und Muster die Daten aufweisen.

Die Daten sollten außerdem fürs Training aufbereitet und in einen Trainings- und Testdatensatz aufgeteilt werden. Der Trainingsdatensatz wird zum Trainieren des Algorithmus verwendet, während der Testdatensatz zum Testen und Anpassen des, vom ausgewählten Algorithmus erstellten Modells, verwendet wird.

2. Auswahl des Machine Learning Algorithmus

Je nachdem um was für Daten es sich handelt, also *labeled* oder *unlabeled data*, und was für ein Problem gelöst werden muss, kann man zwischen verschiedenen Machine Learning Algorithmen wählen, auf die näher in Kapitel 9.5.2 eingegangen wird.

³Zu Deutsch „Maschinelles Lernen“

3. Training des Machine Learning Algorithmus

Das Trainieren des Machine Learning Algorithmus ist ein meist mehrmals durchzuführender Prozess, der daraus besteht, den Algorithmus mit den Werten der Testdaten zu füttern, die Ausgabe des Algorithmus mit der erwarteten Ausgabe zu vergleichen und, bei nicht zufriedenstellenden Ergebnissen, die Parameter des Algorithmus dementsprechend anzupassen.

Dieser Vorgang wird solange wiederholt, bis die Genauigkeit der Ergebnisse des Algorithmus, für das Lösen des Problems, ausreichend ist.

Das Ergebnis dieses Trainings ist das fertige Modell, das nun im realen Betrieb eingesetzt werden kann.

4. Einsatz und Verbesserung des Modells

Optimalerweise wird das Modell nun in der realen Welt eingesetzt und mit neuen Daten gefüttert. Nach einer gewissen Zeit des Einsatzes, kann man das Modell mit den neu erhaltenen Daten weiter trainieren und somit die Qualität der Ergebnisse verbessern.

(SM:Web23, vgl. ibm.com 01.02.2021)

Aufbereitung der Daten

Machine Learning Algorithmen können nur mit numerischen Daten arbeiten. Da im Gebiet des Natural Language Processings allerdings mit Freitexten gearbeitet wird, müssen diese Textdaten in eine numerische Darstellung der Daten umgewandelt werden, bevor sie dem Machine Learning Algorithmus gefüttert werden können. (SM:Web33, vgl. Ameisen 03.02.2021)

Die meisten Verfahren, die sich dies zum Ziel setzen, folgen dem Prinzip des *Word Embeddings*⁴. Beim *Word Embedding* geht es darum, Textdaten in eine Vektorform aus numerischen Werten zu bringen. (SM:Web34, vgl. Khandelwal 03.02.2021)

Für diesen Vorgang gibt es verschiedene Verfahren:

Bag of Words/One-Hot Encoding

Angenommen man hat die verschiedenen Sätze eines Textes in folgender Form gegeben:

⁴Zu Deutsch etwa „Worteinbettung“

Satz
<i>Mary is hungry for apples.</i>
<i>John is happy he is not hungry for apples.</i>

Um diese beiden Beispielsätze in eine numerische Darstellung zu bringen, kann ein *One-Hot Encoding* (ger.: One-Hot-Kodierung) durchgeführt werden. Dabei werden die Sätze mithilfe von Tokenisierung in ihre einzelnen Wörter zerlegt und die einzigartigen Wörter aus beiden Sätzen werden in eine Listenform gebracht, die in diesem Beispiel folgendermaßen aussieht:

[Mary, is, hungry, happy, for, apples, not, John, he]

Diese Liste nimmt keine Rücksicht auf die Reihenfolge der Wörter, sondern stellt lediglich die Menge aller einzigartigen Wörter dar und wird deshalb auch *Bag of Words* genannt. (SM:Web33, vgl. Ameisen 03.02.2021)

Die beiden Sätze werden nun mit dieser Liste abgeglichen und es wird für jeden Satz eine Liste aus Zahlenwerten erstellt, wobei jede Zahl die Anzahl darstellt, wie oft das gegebene Wort aus dem *Bag of Words* in dem Satz vorkommt:

Die Verarbeitung der Beispielsätze sieht dabei folgendermaßen aus:



Abbildung 9.5: One-Hot Encoding
Quelle: (SM:Web33, vgl. Ameisen 03.02.2021)

(SM:Web33, vgl. Ameisen 03.02.2021)

TF-IDF

Durch dieses Verfahren kann man das, durch One-Hot Encoding erstellte, Bag of Words Modell um einen sogenannten *TF-IDF score* (Term Frequency, Inverse Document Frequency) erweitern. (SM:Web33, vgl. Ameisen 03.02.2021)

Der *TF-IDF score* wird für jedes Wort erstellt und bewertet wie selten bzw. wie häufig ein Wort in den Eingabedaten vorkommt. Wenn man als Eingabedaten mehrere Textdokumente gegeben hat, kann der *TF-IDF score* der einzelnen Wörter aus der Multiplikation folgender zwei Größen berechnet werden:

- **Term Frequency:**

Diese Größe gibt an, wie oft das Wort innerhalb des gegebenen Dokuments vorkommt. Dabei kann auf einfachste Weise gezählt werden, wie oft das Wort im Dokument vorkommt. Man kann die *Term Frequency* aber auch anpassen, indem man die Dokumentlänge oder die Häufigkeit des am öftesten vorkommenden Wortes im Dokument berücksichtigt. (SM:Web35, vgl. Stecanella 03.02.2021)

- **Inverse Document Frequency:**

Diese Größe gibt an, wie oft das Wort auf alle Dokumente verteilt vorkommt. Sie wird berechnet, indem man die Anzahl an gegebenen Dokumenten durch die Anzahl der Dokumente dividiert, in denen das Wort zumindest einmal vorkommt. (SM:Web36, vgl. Nguyen 03.02.2021)

Word2Vec

Word2Vec ist ein von Google entwickelter Algorithmus, mit dem man Texte in eine Vektorform bringen kann. Der Algorithmus basiert auf der Verteilungshypothese (*distributional hypothesis*), die besagt, dass Wörter, die oft die selben benachbarten Wörter haben, wahrscheinlich die selbe semantische Bedeutung haben. (SM:Web34, vgl. Khandelwal 26.02.2021)

Mithilfe von Word2Vec trainierte Modelle können, wenn sie mit genügend Daten trainiert wurden, sehr genaue Vorhersagen über die Verhältnisse zwischen verschiedenen Wörtern, aufgrund ihrer Vektordarstellungen, treffen. (SM:Web43, vgl. Nicholson 26.02.2021)

Die Vektordarstellungen der Verhältnisse *Männlich-Weiblich*, *Zeitformen des Verbs* und *Land-Hauptstadt* könnten folgendermaßen aussehen:

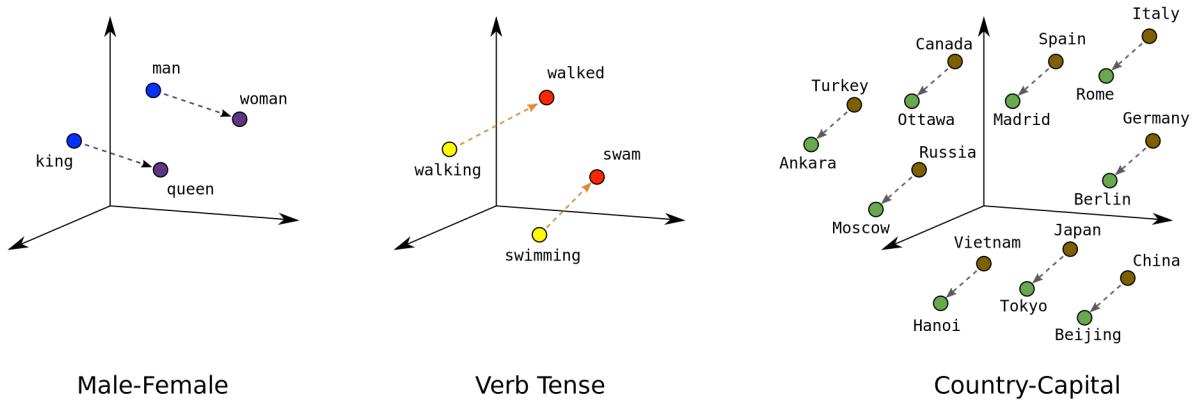


Abbildung 9.6: Vektordarstellungen mithilfe von Word2Vec
Quelle: (SM:Web44, vgl. developers.google.com 26.02.2021)

Machine Learning Algorithmen

Supervised Learning

Beim Supervised Learning füttert man den Machine Learning Algorithmus mit Texten, in denen die vorherzusagende Größe (z. B. der POS-Tag eines Wortes, die Kategorie eines Zeitungsartikels, usw.) im Vorhinein gekennzeichnet wurde (*labeled data*). (SM:Web27, vgl. Barba 02.02.2021)

Die Probleme, die mithilfe von Supervised Learning Algorithmen gelöst werden können, kann man grob in die folgenden Kategorien einteilen:

- **Klassifikationsprobleme:**

Von Klassifikationsproblemen ist die Rede, wenn es sich bei der vorherzusagenden Größe um eine Kategorie, ein Genre, usw. handelt. Ein Beispiel dafür im Gebiet des Natural Language Processings wäre die Vorhersage der Kategorie eines Zeitungsartikels („Sport“ oder „Politik“). (SM:Web25, vgl. Brownlee 01.02.2021)

Konkreter **Algorithmus** zur Lösung solcher Probleme:

- **Logistische Regression:**

Die logistische Regression verdankt ihren Namen der logistischen Funktion

$$f(x) = \frac{1}{1 + e^{-x}}$$

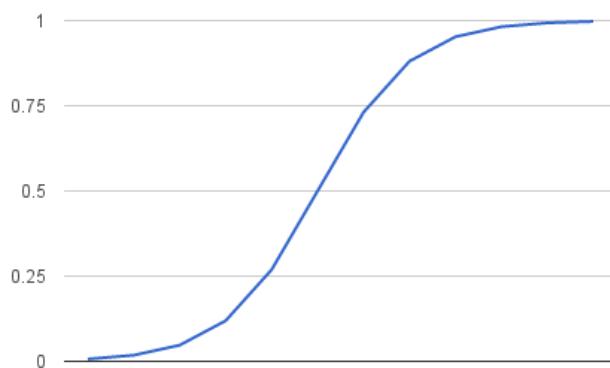


Abbildung 9.7: Logistische Funktion
Quelle: (SM:Web29, vgl. Brownlee 02.02.2021)

Dieser Algorithmus eignet sich hervorragend für Klassifikationsprobleme, bei denen die vorherzusagende Größe nur zwei Werte annehmen kann (z. B. „erkrankt“ oder „nicht erkrankt“, „positive Aussage“ oder „negative

Aussage“), da das Ergebnis der logistischen Funktion nur Werte im Bereich $[0, 1]$ annehmen kann. (SM:Web30, vgl. Chris 02.02.2021)

Angenommen man will eine Vorhersage y über eine binäre Größe aufgrund von zwei Eingabedaten x_1 und x_2 machen, dann kann folgende logistische Funktion aufgestellt werden:

$$y = \frac{e^{b_0+b_1 \cdot x_1+b_2 \cdot x_2}}{1 + e^{b_0+b_1 \cdot x_1+b_2 \cdot x_2}}$$

Die Koeffizienten b_1 und b_2 stellen die Gewichtungen für die jeweiligen Eingabedaten x_1 und x_2 dar, während b_0 eine generelle Gewichtung ist. Diese Koeffizienten werden durch das Trainieren des Algorithmus bestimmt.

(SM:Web29, vgl. Brownlee 02.02.2021)

Da die vorherzusagende Größe nur zwei Werte annehmen kann, die durch die logistische Regression als 0 oder 1 dargestellt werden, muss bei Ergebnissen der logistischen Funktion, die zwischen 0 und 1 liegen, entschieden werden, welche Vorhersage getroffen wird. Dies wird oft mithilfe eines *Threshold Values* (ger.: Schwellenwert) umgesetzt:

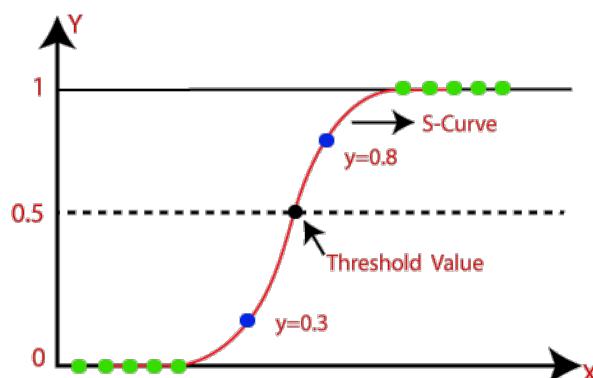


Abbildung 9.8: Veranschaulichung des *Threshold Values*
Quelle: (SM:Web31, vgl. javatpoint.com 02.02.2021)

In diesem Fall sagt die logistische Regression für alle Ergebnisse im Bereich $[0.5, 1]$ den Wert 1 voraus. Für Ergebnisse kleiner als 0.5 wird 0 vorhergesagt.

- **Regressionsprobleme:**

Man spricht von Regressionsproblemen, wenn die vorherzusagende Größe ein fortlaufender numerischer Wert sein soll. Beispiele für Regressionsprobleme wären die Vorhersage des Preises eines Autos oder die Vorhersage der Verbreitung eines Virus in einer bestimmten Region. (SM:Web26, vgl. monkeylearn.com 01.02.2021)

Konkreter **Algorithmus** zur Lösung solcher Probleme:

– **Lineare Regression:**

Bei der linearen Regression handelt es sich um einen Machine Learning Algorithmus, mit dem man einen Schätzwert für eine vorherzusagende Größe, aufgrund von numerischen Eingabedaten, berechnen kann. Das Ziel der linearen Regression ist, die ideale lineare Annäherungsfunktion (*Ausgleichsgerade* (SM:Article11, vgl. Carl-Engler-Schule 02.02.2021)) für die vorherzusagende Größe y und allen n Eingabewerten x_1, x_2, \dots, x_n zu bestimmen. (SM:Web32, vgl. Gandhi 02.02.2021)

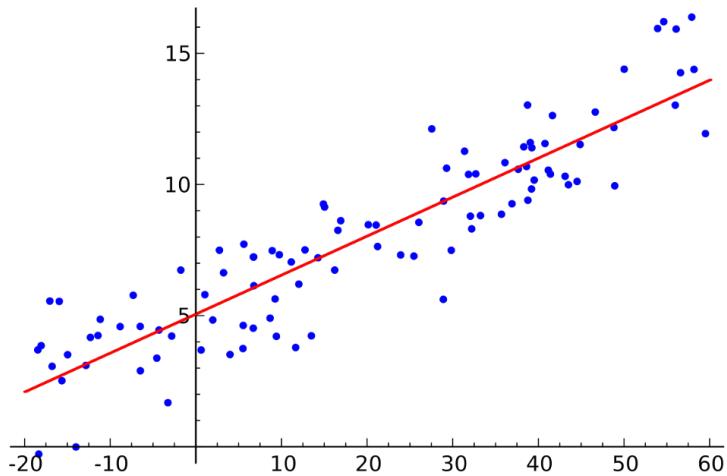


Abbildung 9.9: Ausgleichsgerade (rot) der geg. Daten (blau)
Quelle: (SM:Web32, vgl. Gandhi 02.02.2021)

Diese Ausgleichsgerade könnte durch folgende Funktion dargestellt werden:

$$y = a_0 + a_1 \cdot x$$

Die Variable y stellt den vorherzusagenden Schätzwert dar, während x ein konkreter Eingabewert ist, für den die Vorhersage getroffen werden soll. Die Koeffizienten a_0 und a_1 sind fixe Konstanten, die durch das Trainieren des Algorithmus mit gegebenen Trainingsdaten bestimmt werden.

Diese Koeffizienten sollen durch das Training so bestimmt werden, dass die Summe aller Quadrate der Fehler (Differenz zwischen dem Funktionswert y und der tatsächlichen Größe eines Eingabewerts) möglichst klein ist:

$$\text{minimize} \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

(SM:Web32, vgl. Gandhi 02.02.2021)

Beispiel für den Einsatz von Supervised Learning

Supervised Learning im Bereich des Natural Language Processings lässt sich durch folgendes Beispiel demonstrieren: Ein von *Figure Eight*⁵ bereitgestellter Datensatz, „Disasters on Social Media“, beinhaltet ungefähr 10000 Tweets, die über eine Naturkatastrophe berichten könnten.

Ob ein Tweet über eine Katastrophe berichtet oder nicht, wurde bereits ausgearbeitet. Deswegen handelt es sich bei diesen Daten um *labeled data*. Tweets die von Katastrophen berichten wurden mit dem Wort „Relevant“ markiert, alle anderen mit „Not Relevant“ bzw. „Can't Decide“.

Mithilfe eines Supervised Learning Algorithmus soll ein Modell trainiert werden, das vorhersagen kann, ob ein Tweet von einer Naturkatastrophe berichtet oder nicht. Dieses Modell könnte in weiterer Folge für die Entwicklung einer Anwendung genutzt werden, die zum Beispiel Rettungskräfte alarmiert, sobald über eine Naturkatastrophe getweeted wird.

1. Datenaufbereitung:

Damit die Qualität des Modells nach dem Training so hoch wie möglich ist, sollten die Texte der Tweets für das Training vorbereitet werden. Dazu werden irrelevante Zeichen entfernt, der Text wird in eine kleingeschriebene Form gebracht und anschließend in seine einzelnen Wörter geteilt.

Vorher:	“#RockyFire Update => California Hwy. 20 closed in both directions due to Lake County fire - #CAfire #wildfires”
Nachher:	“[rockyfire, update, california, hwy, 20, closed, in, both, directions, due, to, lake, county, fire, cafire, wildfires]”

2. Daten in ihre Vektorform bringen:

Die vorbereiteten Texte werden nun mithilfe eines vortrainierten Word2Vec-Modells in eine Vektorform gebracht. Der Vorteil von Word2Vec in diesem Fall ist, dass die semantische Bedeutung in die Vektorform miteinbezogen wird, wodurch bessere Ergebnisse erzielt werden können.

Projiziert man die Vektoren nun auf den 2-dimensionalen Raum, kann man erkennen, dass eine gute Differenzierung zwischen Texten die Naturkatastrophen beschreiben und Texten die keine Katastrophen beschreiben, möglich ist:

⁵Unternehmen für maschinelles Lernen und künstliche Intelligenz

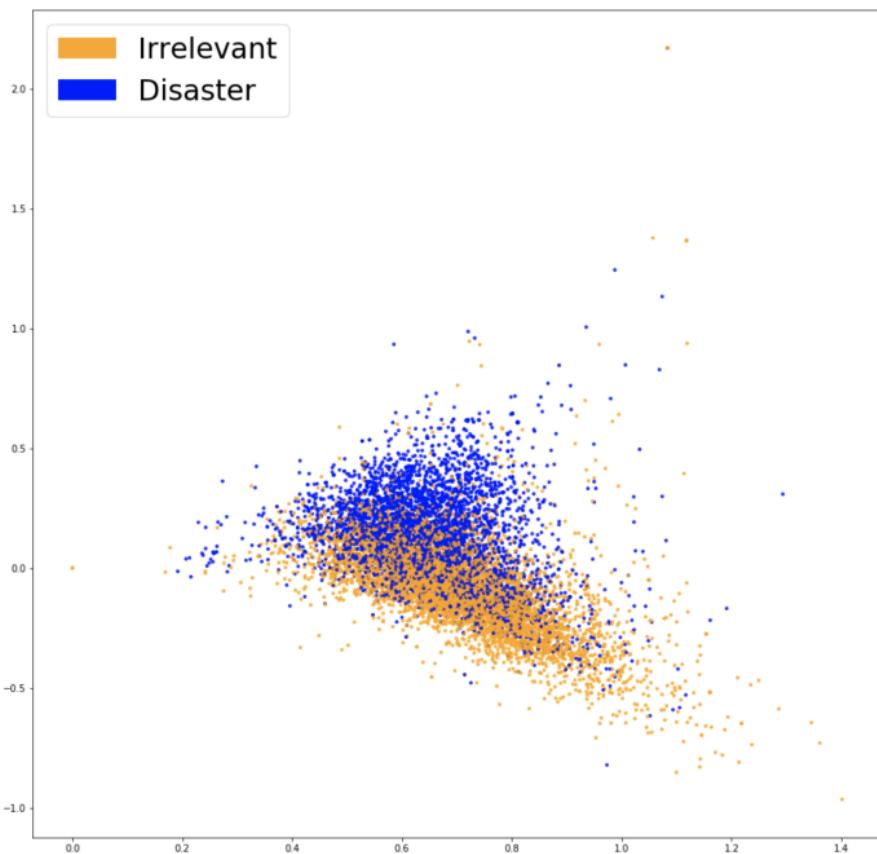


Abbildung 9.10: Visualisierung der Vektorformen
Quelle: (SM:Web33, vgl. Ameisen 15.03.2021)

3. Logistische Regression:

Mithilfe einer logistischen Regression kann nun das Modell trainiert werden. Dazu wird der Datensatz in einen Trainings- und einen Testdatensatz aufgeteilt. Der Trainingsdatensatz wird für das Training des Modells verwendet, während der Testdatensatz zur Bestimmung der Qualität des Modells verwendet wird.

```

1 clf_w2v = LogisticRegression(C=30.0, class_weight='balanced', solver='
  newton-cg', multi_class='multinomial', random_state=40)
2 clf_w2v.fit(X_train_word2vec, y_train_word2vec)
3 y_predicted_word2vec = clf_w2v.predict(X_test_word2vec)
4 accuracy_word2vec, precision_word2vec, recall_word2vec, f1_word2vec =
  get_metrics(y_test_word2vec, y_predicted_word2vec)
5 print("accuracy = %.3f, precision = %.3f, recall = %.3f, f1 = %.3f" %
  (accuracy_word2vec, precision_word2vec, recall_word2vec,
  f1_word2vec))

```

Listing 9.5: Logistische Regression des Datensatzes

accuracy = 0.772, precision = 0.772, recall = 0.772, f1 = 0.772

Das Modell hat eine Genauigkeit von etwa 77%. Das bedeutet, dass etwa 77% der Testdaten vom Modell richtig vorhergesagt wurden.

(SM:Web50, vgl. Ameisen 15.03.2021)

(SM:Web33, vgl. Ameisen 15.03.2021)

Unsupervised Learning

Bei Unsupervised Learning Algorithmen trainiert man ein Modell mithilfe von Texten, in denen die vorherzusagende Größe noch nicht gekennzeichnet wurde (*unlabeled data*). Das Ziel solcher Algorithmen ist das eigenständige Erkennen von Mustern und Zusammenhängen in Texten. (SM:Web58, vgl. Mishra 19.03.2021)

Wenn es um Probleme geht, die mithilfe von Unsupervised Learning Algorithmen gelöst werden können, spricht man meist von Folgendem:

- **Clustering-Probleme:**

Bei Clustering-Problemen geht es darum, Daten in ähnliche Gruppen, auch *Cluster* genannt, einzusortieren. Der Unterschied zu Klassifikationsproblemen (siehe Kapitel 9.5.2) liegt darin, dass man kein Vorwissen über die Daten hat. Der Unsupervised Learning Algorithmus operiert rein auf Ähnlichkeiten zwischen den verschiedenen Daten. (SM:Web45, vgl. Semmelmann 13.03.2021)

Im Gebiet des Natural Language Processings wird Clustering zum Beispiel dazu verwendet, Texte oder Beiträge, anhand ihres Titels, in verschiedene Kategorien einzurichten, ohne zuvor zu wissen, um was es genau in den Texten oder Beiträgen geht. (SM:Web46, vgl. Wilentz 13.03.2021)

Konkreter Clustering-**Algorithmus**:

- **k-Means-Algorithmus:**

Der k-Means-Algorithmus ist dazu in der Lage, Daten im n-dimensionalen Raum in k Cluster einzurichten. Die Anzahl der Cluster k muss dem Algorithmus dabei im Vorhinein bekanntgegeben werden. (SM:Web47, vgl. Luber 13.03.2021)

Der Algorithmus arbeitet dabei folgende Schritte ab (Veranschaulichung im 2-dimensionalen Raum):

1. Auswahl von k Punkten als Anfangszentren der Cluster.
2. Zuweisung aller Daten zu dem Cluster, zu dem sie den geringsten Abstand haben.
3. Neuberechnung der Clusterzentren.

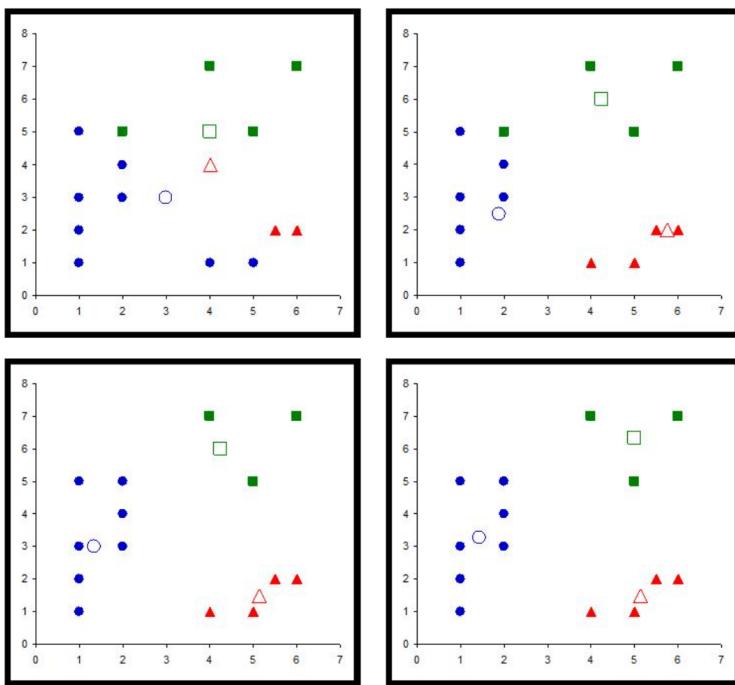


Abbildung 9.11: Veranschaulichung des k-Means-Algorithmus
Quelle: (SM:Web48, vgl. TU München 13.03.2021)

Die Schritte 2 und 3 werden so lange wiederholt bis sich die Positionen der Clusterzentren nicht mehr ändern.

(SM:Web48, vgl. TU München 13.03.2021)

Die Qualität des Clusterings kann dabei durch den *Silhouetten-Koeffizienten* angegeben werden. Je dichter die einzelnen Cluster sind und desto weiter sie von den jeweils anderen Clustern entfernt sind, desto größer ist der Silhouetten-Koeffizient (SM:Web46, vgl. Wilentz 14.03.2021). Dieser ist dabei unabhängig von der Anzahl der Cluster k . (SM:Article13, vgl. LMU München 14.03.2021)

Der Silhouetten-Koeffizient lässt sich folgendermaßen berechnen: Wenn $a(o)$ der Abstand eines Datenobjekts zum Zentrum seines Clusters und $b(o)$ der Abstand eines Datenobjekts zum Zentrum seines zweitnächstgelegenen Clusters ist, ergibt sich die Silhouette $s(o)$ des Datenobjekts aus folgender Formel:

$$s(o) = \frac{b(o) - a(o)}{\max\{a(o), b(o)\}}$$

Der Silhouetten-Koeffizient s_C eines Clusterings stellt den Durchschnitt der Silhouetten aller Datenobjekte dar.

(SM:Article13, vgl. LMU München 14.03.2021)

Beispiel für den Einsatz von Unsupervised Learning

Mit folgendem Beispiel kann man Unsupervised Learning im Gebiet des Natural Language Processings demonstrieren: Im Subreddit *r/food* werden hübsche Bilder von Mahlzeiten gepostet. Jeder Beitrag in diesem Subreddit hat einen Titel. Anhand der Wörter in diesen Titeln sollen Strukturen in diesem Subreddit erkannt werden.

1. Datenbeschaffung:

Mithilfe der *pushshift API* werden die Titel der Top 1000 Beiträge pro Monat der letzten 48 Monate beschafft. Somit hat man nun einen Datensatz aus 48000 Titeln.

2. Datenaufbereitung:

Damit der Machine Learning Algorithmus besser mit den Daten arbeiten kann, müssen die Daten aufbereitet werden. Dazu werden die Titel kleingeschrieben und Satzzeichen, Emojis und Zahlen werden entfernt.

Vorher: “1/4 hotdog wrapped in bacon w/Mac & cheese, BBQ sauce[OC][4023X3024]”

Nachher: “hotdog wrapped bacon w mac cheese bbq sauce”

3. Daten in ihre Vektorform bringen:

Mithilfe des *TF-IDF*-Verfahrens (siehe Kapitel 9.5.2) werden die Titel in eine Vektorform gebracht. Dabei wird jedem Wort im Titel ein Zahlenwert zugewiesen.

Dem Wort „bacon“ wird zum Beispiel ein niedrigerer Wert zugewiesen als dem Wort „sashimi“, da das Wort „bacon“ häufiger in Titeln vorkommt als das Wort „sashimi“. Das heißt, dass das Wort „sashimi“ für die Differenzierung der Bedeutungen von Titeln wichtiger ist als das Wort „bacon“.

4. Clustering:

Mithilfe des k-Means-Algorithmus werden die Daten nun in k Kategorien eingeteilt. Mithilfe der Ellbogenmethode kann man den optimalen Wert für k bestimmen. Dabei stellt man die Silhouetten-Koeffizienten für k in einem bestimmten Bereich, zum Beispiel von 2 bis 15, grafisch dar und wählt dann das k bei dem sich der Silhouetten-Koeffizient im „Ellbogen der Kurve“ befindet.

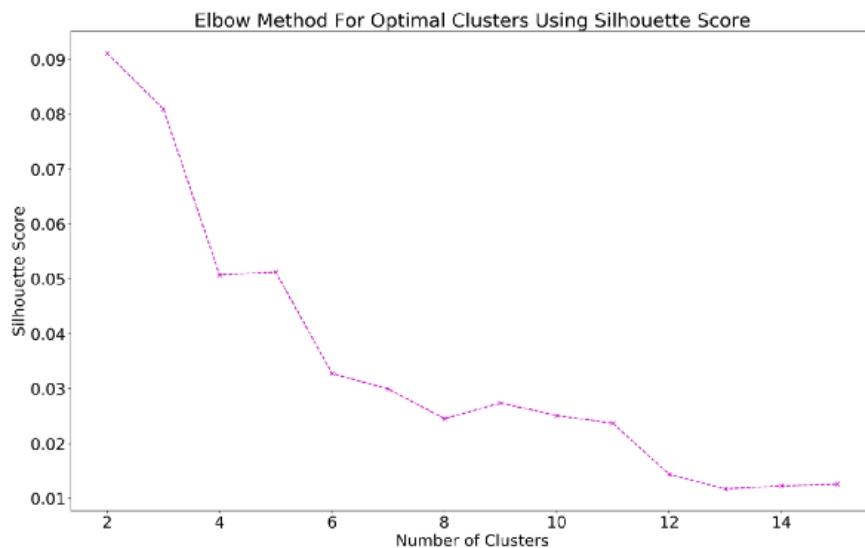


Abbildung 9.12: Ellbogenmethode
Quelle: (SM:Web46, vgl. Wilentz 14.03.2021)

In diesem Fall wird für k der optimale Wert 6 gewählt.

Nun kann man mithilfe des k-Means-Algorithmus die Beiträge anhand ihrer Titel in 6 Kategorien einteilen.

5. Auswertung der Ergebnisse:

Um die Ergebnisse auswerten zu können, kann man sich zum Beispiel für jede Kategorie die 10 wichtigsten Wörter anzeigen lassen. Dies kann folgendermaßen aussehen:

```
display_topics(nmf_model_tf_idf, tf_idf.get_feature_names(), 10)

Topic 0
cheese, bacon, mac, burger, grilled, egg, wrapped, fries, cheddar, cream

Topic 1
pizza, pepperoni, crust, deep, cast, mozzarella, iron, bread, margherita, oven

Topic 2
chicken, fried, rice, waffles, curry, wings, spicy, salad, sandwich, soup

Topic 3
cake, chocolate, cream, ice, cookies, chip, butter, pie, strawberry, caramel

Topic 4
breakfast, sandwich, egg, eggs, morning, toast, english, healthy, avocado, bacon

Topic 5
steak, pork, beef, garlic, dinner, potatoes, sauce, salad, eggs, smoked
```

Abbildung 9.13: 10 wichtigsten Wörter pro Kategorie
Quelle: (SM:Web49, vgl. Wilentz 14.03.2021)

Anhand dieser 10 wichtigsten Wörter, kann man nun Vermutungen aufstellen, um was es sich bei den Kategorien handeln könnte. In diesem Fall kann man vermuten, dass es sich bei der Kategorie *Topic 0* um Beiträge über Burger, bei *Topic 1* um Beiträge über Pizzen, usw. handelt.

(SM:Web46, vgl. Wilentz 14.03.2021)

9.6 Natural Language Processing und künstliche neuronale Netze

9.6.1 Was ist ein künstliches neuronales Netz?

Ein künstliches neuronales Netz ist ein, dem menschlichen Gehirn nachempfundenes, Machine Learning Modell. Es besteht aus vielen künstlichen Neuronen, durch deren Zusammenarbeit Probleme des Natural Language Processings, der Statistik und anderer Gebiete gelöst werden können. Bevor man mit einem künstlichen neuronalen Netz ein Problem lösen kann, muss es, wie jedes andere Machine Learning Modell, mit gegebenen Daten trainiert werden. (SM:Web37, vgl. Luber 04.02.2021)

Künstliches Neuron

Um die Funktion eines künstlichen Neurons in einem künstlichen neuronalen Netz zu verstehen, muss man sich zuerst den Aufbau eines Neurons im menschlichen Gehirn ansehen:

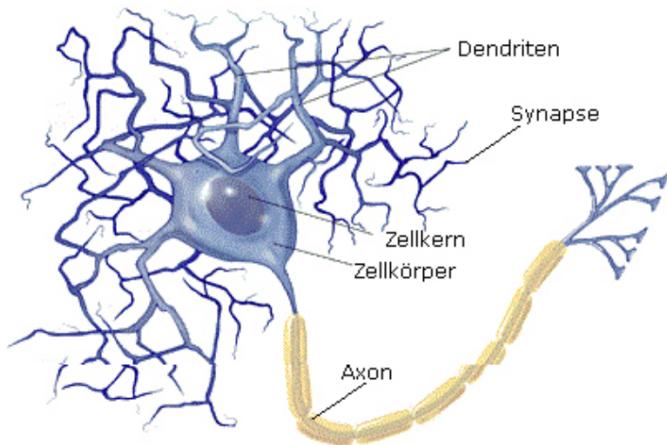


Abbildung 9.14: Das biologische Neuron
Quelle: (SM:Article12, vgl. Ruland 04.02.2021)

Ein biologisches Neuron besteht aus einem Zellkörper mit Zellkern, von diesem Zellkörper aus verlaufen die Dendriten. An den Enden der Dendriten befinden sich die **Synapsen**. Vom Zellkörper aus verläuft noch eine dickere Faser, das **Axon**. (SM:Article12, vgl. Ruland 04.02.2021)

Die Synapsen und das Axon sind bei einem Neuron für die „Datenübertragung“ verant-

wortlich. Die Synapsen empfangen dabei elektrische Signale von anderen Neuronen und wenn diese Signale stark genug sind, sendet das Neuron ein eigenes Signal über das Axon. Dabei ist die Signalübertragung der Synapsen nicht bei jedem Neuron gleich, sondern man unterscheidet zwischen hemmenden und erregenden Synapsen. (SM:Article12, vgl. Ruland 04.02.2021)

Mit diesem Wissen kann man nun ein künstliches Neuron modellieren:

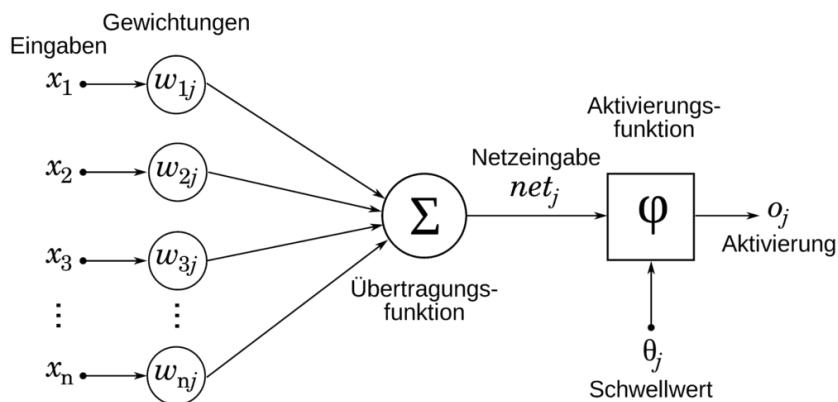


Abbildung 9.15: Aufbau eines künstlichen Neurons
Quelle: (SM:Web38, vgl. Wuttke 04.02.2021)

Bei einem künstlichen Neuron stellen die n Eingabewerte x_1, x_2, \dots, x_n die Synapsen eines biologischen Neurons dar. Diese Eingabewerte werden nun mit ihren jeweiligen Gewichtungen multipliziert und im „Zellkern“ aufsummiert. Das Ziel des Trainings eines künstlichen neuronalen Netzes ist die optimalen Gewichtungen für die Eingabewerte der einzelnen künstlichen Neuronen zu bestimmen. (SM:Article12, vgl. Ruland 04.02.2021)

Die Summe der gewichteten Eingaben wird auch Netzeingabe net_j genannt und dient als Eingabewert für die Aktivierungsfunktion des künstlichen Neurons. Unter Berücksichtigung eines Schwellenwerts θ_j bestimmt die Aktivierungsfunktion was für ein Signal das künstliche Neuron über seine Ausgabeschicht (seinem „Axon“), an andere künstliche Neuronen, weiterschicken soll. (SM:Web38, vgl. Wuttke 04.02.2021)

Aufbau eines künstlichen neuronalen Netzes

Aus vielen solcher künstlichen Neuronen kann nun ein künstliches neuronales Netz aufgebaut werden. Der grundsätzliche Aufbau sieht dabei folgendermaßen aus:

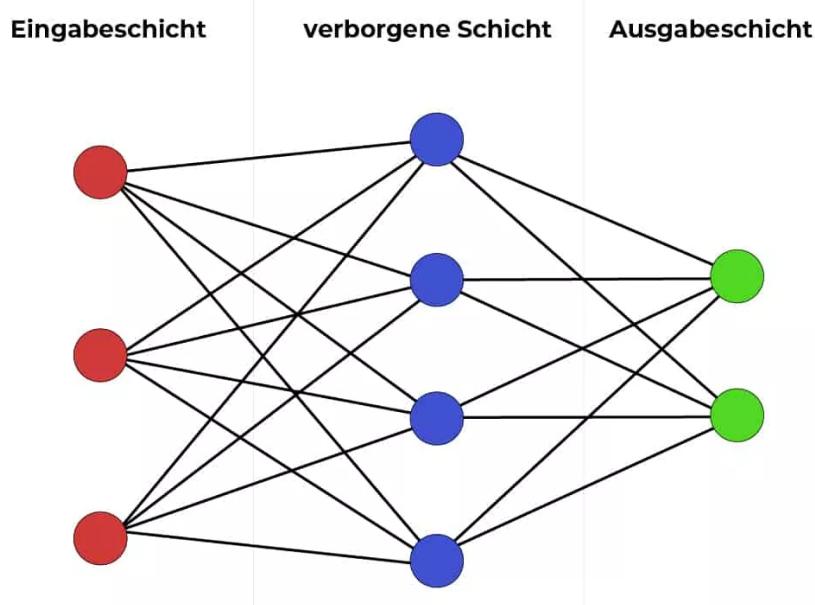


Abbildung 9.16: Grundsätzlicher Aufbau eines künstlichen neuronalen Netzes
Quelle: (SM:Web38, vgl. Wuttke 04.02.2021)

Ein künstliches neuronales Netz besteht dabei immer aus folgenden Schichten von künstlichen Neuronen:

- Einer **Eingabeschicht**
- Einer oder mehrerer **verborgener Schichten**
- Einer **Ausgabeschicht**

Die Eingabeschicht ist dafür zuständig, das neuronale Netz mit den Eingabedaten in gewichteter Form zu versorgen. Die Neuronen der Eingabeschicht geben die eingegebenen Daten weiter an die erste bzw. eventuell einzige verborgene Schicht weiter. Auf die erste verborgene Schicht können beliebig viele weitere verborgene Schichten folgen. (SM:Web38, vgl. Wuttke 19.03.2021)

Innerhalb der verborgenen Schichten werden die Informationen über weitere gewichtete Verbindungen bis zur Ausgabeschicht weitergereicht. Wie genau die Verarbeitung der

Informationen innerhalb der verborgenen Schichten abläuft, ist dabei nicht ersichtlich. Die Ausgabeschicht liefert zu guter Letzt das Ergebnis der Berechnung des neuronalen Netzes. (SM:Web38, vgl. Wuttke 19.03.2021)

Wie diese Schichten miteinander verbunden werden, kann auf verschiedene Weisen geschehen (siehe Kapitel 9.6.2).

9.6.2 Arten von künstlichen neuronalen Netzen

Recurrent Neuronal Networks

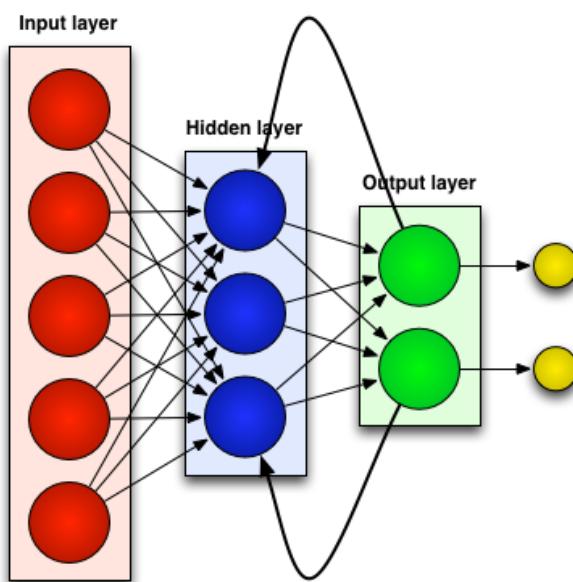


Abbildung 9.17: Recurrent Neural Network
Quelle: (SM:Web39, vgl. Roell 05.02.2021)

Bei Recurrent Neural Networks (zu Deutsch etwa „rekurrente⁶ neuronale Netze“) handelt es sich um eine Art von künstlichen neuronalen Netzen, die im Gebiet des Natural Language Processing häufig dazu verwendet werden, Autovervollständigung (Vorhersage des nächsten Wortes) umzusetzen. (SM:Web40, vgl. Sharma 05.02.2021)

Das Besondere an Recurrent Neural Networks ist, dass sie einen „internen Speicher“ besitzen, wodurch sie sich bisher eingegebene Daten merken und für weitere Berechnungen verwenden können (SM:Web51, vgl. ai-united.de 16.03.2021).

⁶sich (beständig) wiederholend, immer wiederkehrend

Dies kann durch verschiedene Rückkopplungen realisiert werden:

- **Direkte Rückkopplungen:** Dabei speisen die Ausgangsdaten eines Neurons den Eingang des selben Neurons.
- **Indirekte Rückkopplungen:** Dabei werden die Ausgangsdaten eines Neurons als Eingangsdaten eines Neurons einer vorherigen Schicht verwendet.
- **Seitliche Rückkopplungen:** Hier stellen die Ausgangsdaten eines Neurons die Eingangsdaten eines Neurons der selben Schicht dar.
- **Vollständige Rückkopplungen:** Bei vollständigen Rückkopplungen werden alle Neuronenausgänge als Eingänge aller anderen Neuronen genutzt.

Verglichen mit dem menschlichen Gehirn, lässt sich dieses Konzept der Rückkopplungen mit einem Gedächtnis vergleichen.

(SM:Web52, vgl. Luber 16.03.2021)

Long Short Term Memory

Bei Recurrent Neural Networks besteht ein Problem darin, dass sie weit zurückliegende Informationen nach vielen Durchgängen „vergessen“ bzw. nicht mehr effizient auffinden können. Indem man das Netz durch sogenannte LSTM-Zellen (Long Short Term Memory⁷) ergänzt, lässt sich dieses Problem lösen. (SM:Web52, vgl. Luber 16.03.2021)

Eine LSTM-Zelle muss so intelligent sein, dass sie entscheiden kann, welche Informationen für wie lange gespeichert werden sollen. Außerdem muss sie dazu in der Lage sein, Verbindungen zwischen gespeichertem Wissen und neuen Informationen herzustellen. (SM:Web53, vgl. Luber 17.03.2021)

⁷Zu Deutsch etwa „langes Kurzzeitgedächtnis“

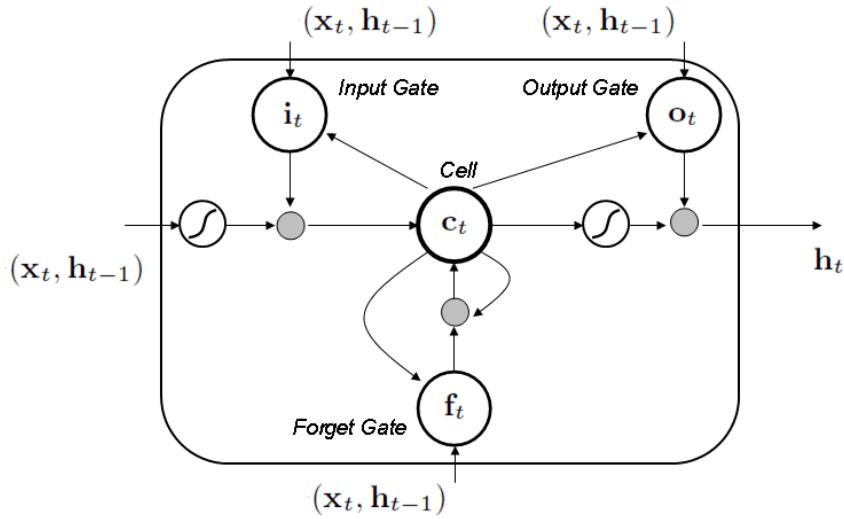


Abbildung 9.18: Veranschaulichung einer LSTM-Zelle
Quelle: (SM:Web54, vgl. Hoory 17.03.2021)

Um diese Funktionalitäten umsetzen zu können, wird eine LSTM-Zelle aus folgenden Teilen aufgebaut:

- **Input Gate:** Das Input Gate (ger.: Eingangstor) entscheidet, welche Informationen in welchem Umfang in die Zelle gelangen.
- **Forget Gate:** Über das Forget Gate (ger.: Vergessstor) wird entschieden, ob die Informationen gespeichert oder verworfen werden.
- **Output Gate:** Das Output Gate (ger.: Ausgangstor) bestimmt, in welcher Form die Informationen auszugeben sind.
- **Zellinnere:** Im Zellinneren wird die Verknüpfungslogik geregelt und die Informationsflüsse und Speichervorgänge gesteuert.

(SM:Web53, vgl. Luber 17.03.2021)

Convolutional Neural Networks

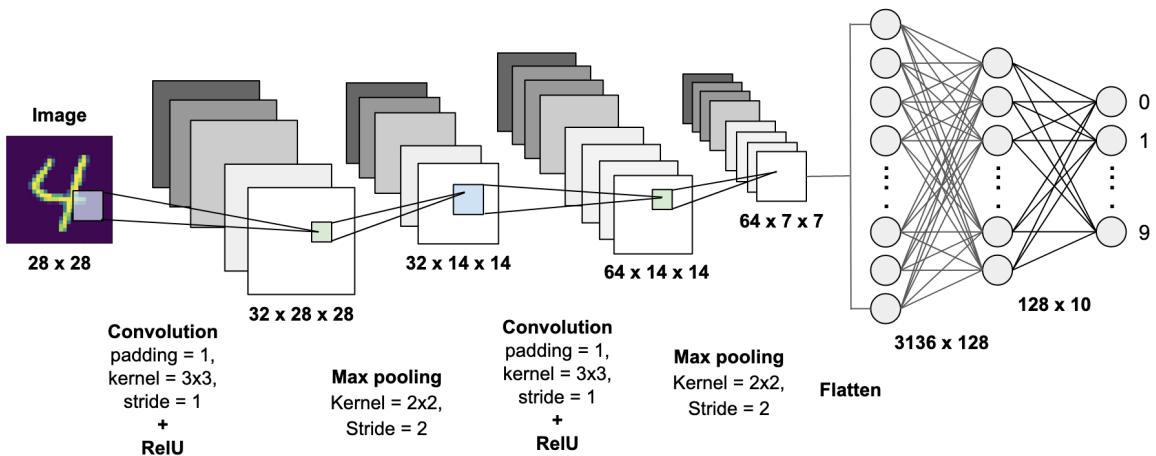


Abbildung 9.19: Convolutional Neural Network
Quelle: (SM:Web41, vgl. Shreyak 05.02.2021)

Convolutional Neural Networks (zu Deutsch etwa „gefaltete neuronale Netze“) sind künstliche neuronale Netze, die sich besonders gut dazu eignen, Bilder zu erkennen. Im Gebiet des Natural Language Processing werden sie zum Beispiel dazu genutzt, geschriebene Schrift zu erkennen. (SM:Web42, vgl. Saha 05.02.2021)

Dabei ist ein Convolutional Neural Network aus folgenden Schichten aufgebaut:

- einer oder mehrerer **Convolutional-Schichten**
- einer oder mehrerer **Pooling-Schichten**
- einer **vollständig verknüpften Schicht**

(SM:Web55, vgl. Luber 18.03.2021)

Eine Convolutional-Schicht hat die Aufgabe, Merkmale in den Eingabedaten zu bestimmen und zu erfassen. Diese erkannten Merkmale werden an eine Pooling-Schicht weitergeleitet, die die wichtigsten Merkmale herausfiltert und überflüssige Informationen verwirft. Dadurch wird die Datenmenge reduziert. Nun können weitere Paare aus Convolutional-Schicht und Pooling-Schicht folgen. (SM:Web55, vgl. Luber 18.03.2021)

Am Ende des Convolutional Neural Networks befindet sich eine vollständig verknüpfte Schicht. Diese Schicht stellt eine gewöhnliche neuronale Netzstruktur dar, in der alle

Ausgänge der einen Schicht mit allen Eingängen der darauffolgenden Schicht verbunden sind. Die Ausgabeschicht dieser Netzstruktur besitzt genau soviele Neuronen, wie es vorherzusagende Werte gibt. Möchte man zum Beispiel ein Convolutional Neural Network trainieren, dass eine händisch geschriebene Ziffer erkennen soll, hat die Ausgabeschicht dieses Netzes zehn Ausgabeneuronen (für die Ziffern 0 bis 9). (SM:Web56, vgl. Becker 18.03.2021)

Kapitel 10

Analyse von Börsennews im Projekt AI Börse

10.1 Aufgabenstellung

Die, vom Crawler automatisiert erfassten Börsenbriefe, sind als Freitexte in einer PostgreSQL Datenbank abgespeichert. Mithilfe eines Python-Programms sollen aus diesen Texten automatisiert Grenze-Anlaufstelle-Paare der Prognosewerte erfasst werden.

So ein Börsenbrief könnte folgenden Inhalt haben:

Weiter steigende Anleiherenditen machen die Dax-Anleger nervös. Zum gestrigen Handelsauftakt notierte der Dax noch über der 14.000er Marke und markierte dabei ein Tageshoch bei 14.051 Punkten. (...) Der deutsche Leitindex ist im frühen Handel an die Unterstützung bei 13.670 gelaufen. Dort konnte er zunächst Halt finden. Fällt er jedoch darunter, dann dürfte noch die markante Unterstützungszone bei 13.615 bis 13.600 angelauft werden. Unterhalb von 13.600 trübt sich das Bild weiter ein, die nächsten Anlaufpunkte liegen dann bei 13.562 und 13.475. Bei 13.475 hätte der Dax eine volle Impulsauffächerung vom Top im H4-Chart abgearbeitet. Händler sollten auch heute ein Auge auf die Anleiherenditen haben, aktuell notieren diese 2,8% im Minus bei 1,472. Kann sich der Dax an der Unterstützung bei 13.670 oder 13.600 stabilisieren, dann dürfte es zu einem Erholungsversuch kommen. Die ersten Widerstände befinden sich bei 13.764, 13.782 und 13.800. Oberhalb der 13.800 könnte der Dax noch die 13.824 und 13.855 anlaufen. Der Widerstand an der 13.855 sollte den Dax zunächst aufhalten, dort könnte er wieder den Rückwärtsgang einlegen. Das Chartbild hellt sich

deutlich auf, wenn der Dax das 61er Retracement bei 13.917 überwindet. Dax Unterstützungen (US): 13.664 – Tagestief 23.02. 13.614 – 61,8% Retracement (14.190 -13.270). 13.600 – 61,8% Extension (Top im daily). 13.562 – 423,6% Extension. 13.475 – 423,6% Extension (Top im H4). Dax Widerstände (WS): 13.782 – Nachthoch. 13.800 – Tief 22.02. 13.855 – vormals US. 13.917 – 61,8% Retracement (H1). 13.912 – nachbörsliches Hoch. 13.988 – 78,6% Retracement.

Quelle: <https://finanzmarktwelt.de>

Die erfassten Prognosewerte sollen folgendermaßen aussehen:

	id integer	grenze double precision	anlaufstelle double precision	link character varying (255)	datum date	uhrzeit character varying (255)
1	3220	13475	13270	https://finanzmarktwelt.de/da...	2021-02-26	09:23:47
2	3219	13500	13475	https://finanzmarktwelt.de/da...	2021-02-26	09:23:47
3	3218	13562	13500	https://finanzmarktwelt.de/da...	2021-02-26	09:23:47
4	3217	13600	13562	https://finanzmarktwelt.de/da...	2021-02-26	09:23:47
5	3216	13614	13600	https://finanzmarktwelt.de/da...	2021-02-26	09:23:47
6	3215	13615	13614	https://finanzmarktwelt.de/da...	2021-02-26	09:23:46
7	3214	13664	13615	https://finanzmarktwelt.de/da...	2021-02-26	09:23:46
8	3213	13670	13664	https://finanzmarktwelt.de/da...	2021-02-26	09:23:46
9	3205	13764	13782	https://finanzmarktwelt.de/da...	2021-02-26	09:23:46
10	3206	13782	13800	https://finanzmarktwelt.de/da...	2021-02-26	09:23:46
11	3207	13800	13855	https://finanzmarktwelt.de/da...	2021-02-26	09:23:46
12	3208	13855	13912	https://finanzmarktwelt.de/da...	2021-02-26	09:23:46
13	3209	13912	13917	https://finanzmarktwelt.de/da...	2021-02-26	09:23:46
14	3210	13917	13988	https://finanzmarktwelt.de/da...	2021-02-26	09:23:46
15	3211	13988	14000	https://finanzmarktwelt.de/da...	2021-02-26	09:23:46
16	3212	14000	14051	https://finanzmarktwelt.de/da...	2021-02-26	09:23:46

Abbildung 10.1: Erfasste Prognosewerte in Tabellenform

In diesem Kapitel wird auf die verschiedenen Ansätze eingegangen, mit denen im Projekt AI Börse versucht wurde, diese Funktionalität umzusetzen.

10.2 Analyse mithilfe von Named-Entity-Recognition

Die Programme, die für diesen Ansatz erstellt wurden, sind auf dem Datenträger unter X:\Programme\named_entity_recognition zu finden.

Die erste Idee war, ein Named-Entity-Recognition-Modell (siehe Kapitel 9.3.2) zu trainieren, dass die Grenzen und Anlaufstellen in den einzelnen Börsenbriefen eigenständig erkennen kann.

10.2.1 Verwendete Technologien

spaCy

Bei spaCy handelt es sich um eine Python-Library, die Funktionalitäten des Natural Language Processings zur Verfügung stellt. Dabei können Texte an eine *Pipeline* übergeben werden, die typischerweise aus mehreren Komponenten besteht und den Text auf verschiedene Arten verarbeiten kann. So eine Pipeline könnte folgendermaßen aussehen:

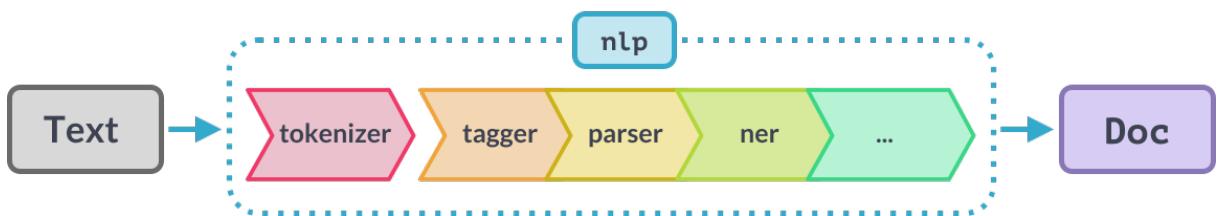


Abbildung 10.2: Typische spaCy Pipeline
Quelle: (SM:Web59, vgl. spaCy Dokumentation 24.03.2021)

Dabei kann man zu vortrainierten Pipelines greifen oder selbst eine Pipeline trainieren. Da es zur automatischen Erkennung von Prognosewerten zum DAX-Kurs keine vortrainierten Pipelines gibt, wurde sich im Projekt AI Börse dazu entschieden, eine eigene Pipeline zu trainieren. Diese soll als einzige Komponente einen *EntityRecognizer* *ner* haben.

Der *EntityRecognizer* in einem spaCy System arbeitet dabei mit *Word Embeddings* (siehe Kapitel 9.5.2) und einem *Convolutional Neural Network* (siehe Kapitel 9.6.2). (SM:Web63, vgl. Bhavani 25.03.2021)

10.2.2 Implementation

Beschaffung der Trainingsdaten

Dazu mussten Trainingsdaten in folgender Form gewonnen werden:

```

1 TRAIN_DATA = [
2   ("<Text>", {
3     "entities": [(anfang1, ende1, "entity1"), (anfang2, ende2, "entity2"),
4                   ...]
5   }),
6   ...
7 ]

```

Listing 10.1: Format der Trainingsdaten

Dabei ist `<Text>` ein beliebiger Text und `entities` eine Liste von *Entities*, die im Text vorkommen. Die Entity `entity1` ist dabei im Text zwischen der Stelle `anfang1` (inklusive) und `ende1` (exklusive) aufzufinden.

Auf diese Weise wurden folgende Trainingsdaten aufgestellt:

```

1 TRAIN_DATA = [
2   ("Kaufpositionen über 13919,00 mit Kurszielen von 14170,00 & 14230,00.",
3    {
4      "entities": [(20, 28, "Grenze"), (48, 56, "Anlaufstelle"), (59, 67,
5                    "Anlaufstelle")]
6    },
7   ("Alternatives Szenario: unter 13919,00 erwarten wir weitere Abwä
8     rtsbewegungen mit 13844,00 & 13728,00 als Kursziele.", {
9      "entities": [(29, 37, "Grenze"), (81, 89, "Anlaufstelle"), (92,
10                  100, "Anlaufstelle")]
11 },
12   ("Unsere Meinung: kaufposition über 13845,00 mit Kurszielen von
13      14060,00 & 14110,00.", {
14      "entities": [(34, 42, "Grenze"), (62, 70, "Anlaufstelle"), (73, 91,
15                  "Anlaufstelle")]
16 },
17   ("Alternatives Szenario: unter 13845,00 erwarten wir weitere Abwä
18     rtsbewegungen mit 13730,00 & 13637,00 als Kursziele.", {
19      "entities": [(29, 37, "Grenze"), (81, 89, "Anlaufstelle"), (92,
20                  100, "Anlaufstelle")]
21 },
22 ...
23   ("Anlaufmarken nach unten bei 13.200, 13.100, 13.000, 12.950, 12.850,
24      12.600 und 12.330 Punkten.", {
25      "entities": [(28, 34, "Grenze"), (36, 42, "Anlaufstelle"), (44, 50,
26                      "Anlaufstelle"), (52, 58, "Anlaufstelle"), (60, 66,
27                      "Anlaufstelle"), (68, 74, "Anlaufstelle"), (79, 85, "Anlaufstelle
28                      ")])
29 ]

```

```

17     }),
18     ("Nach oben bei 13.250, 13.300, 13.350, 13.400 und 13.500 Punkten.", {
19       "entities": [(14, 20, "Grenze"), (22, 28, "Anlaufstelle"), (30, 36,
20         "Anlaufstelle"), (38, 44, "Anlaufstelle"), (49, 55, "Anlaufstelle")]
21   })
]

```

Listing 10.2: Trainingsdaten für die Named Entity Recognition

Training des Modells

Zuerst wird ein leeres Modell für die Sprache Deutsch erzeugt, auf dem weiter aufgebaut werden kann:

```
1 nlp = spacy.blank("de")
```

Mithilfe des Befehls `.add_pipe()` wird die Komponente `ner` zur Pipeline des zuvor erzeugten Modells hinzugefügt:

```
1 nlp.add_pipe("ner", last = True)
```

Durch folgenden Codeabschnitt wird das Modell nun trainiert:

```

1 ner = nlp.get_pipe("ner")
2
3 for _, annotations in TRAIN_DATA:
4     for ent in annotations.get("entities"):
5         ner.add_label(ent[2])
6
7 other_pipes = [pipe for pipe in nlp.pipe_names if pipe != "ner"]
8 with nlp.disable_pipes(*other_pipes):
9     optimizer = nlp.begin_training()
10    for itn in range(n_iter):
11        random.shuffle(TRAIN_DATA)
12        losses = {}
13        examples = []
14        for text, annotations in tqdm(TRAIN_DATA):
15            examples.append(Example.from_dict(nlp.make_doc(text),
16                annotations))
17        nlp.update(
18            examples,
19            drop = 0.5,
20            sgd = optimizer,
21            losses = losses)
22        print(losses)

```

Listing 10.3: Training des NER-Modells

Speichern des Modells

Zuerst muss der Pfad festgelegt werden, in dem das Modell abgespeichert werden soll:

```
1 from pathlib import Path
2 output_dir = Path("/mnt/d/python/boerse/named_entity_recognition/my_model")
```

Nun kann das Modell gespeichert werden:

```
1 if output_dir is not None:
2     output_dir = Path(output_dir)
3     if not output_dir.exists():
4         output_dir.mkdir()
5     nlp.to_disk(output_dir)
6     print("Saved model to", output_dir)
```

Listing 10.4: Speichern des NER-Modells

Laden des Modells

Das Modell kann nun in anderen Python-Programmen verwendet werden, indem man den Befehl `spacy.load()` aufruft:

```
1 nlp = spacy.load("my_model")
```

Listing 10.5: Laden des NER-Modells

Testen des Modells

Nun kann man Testsätze dem Modell übergeben und somit die Qualität der Ergebnisse überprüfen:

```
1 text = ["Kaufposition über 13500 mit Kurszielen von 14000.",
2         "Wenn er über 13000 steigt, steigt er weiter auf 14000"]
3
4 for t in text:
5     doc = nlp(t)
6     print(doc.text, "->", [(ent.text, ent.label_) for ent in doc.ents])
```

Listing 10.6: Testen des NER-Modells

Die Ausgabe dieses Programms sieht folgendermaßen aus:

```
1 <spacy.lang.de.German object at 0x7fb74aed6430>
2 Kaufposition über 13500 mit Kurszielen von 14000. -> [('13500', 'Grenze'),
   ('14000.', 'Anlaufstelle')]
```

³ Wenn er über 13000 steigt, steigt er weiter auf 14000 -> [('13000', 'Grenze'), ('14000', 'Anlaufstelle')]

10.2.3 Problem

Das größte Problem bei diesem Ansatz, stellt die Beschaffung der Trainingsdaten dar. Da man Sätze heraussuchen muss, die sich zum Trainieren des Modells eignen, und in jedem Satz händisch die Stellen abzählen muss, an denen sich die einzelnen Entities befinden, stellt die Beschaffung der Trainingsdaten einen riesigen Zeitaufwand dar, der im Rahmen des Projekts nicht zu bewältigen war.

10.3 Sentiment-Analyse der Börsenbriefe

Die Programme, die für diesen Ansatz erstellt wurden, sind auf dem Datenträger unter X:\Programme\sentiment_analyse zu finden.

Ziel dieses Ansatzes war es, auf die einzelnen Sätze der Börsenbriefe eine Sentiment-Analyse, mithilfe verschiedenster vortrainierter Modelle, durchzuführen. Wenn die Analyse den Satz positiv beurteilt, sollen die numerischen Werte im Satz als aufsteigende Prognosewerte herangezogen werden und wenn die Beurteilung negativ ist, sollen sie als absteigende Prognosewerte betrachtet werden.

10.3.1 Verwendete Technologien

analyze sentiment

Bei analyze_sentiment handelt es sich um eine vortrainierte Pipeline der Python-Library Spark NLP, die darauf trainiert wurde, zu erkennen, ob es sich bei einem englischen Text um eine positive oder negative Aussage handelt. Mit Spark NLP lassen sich, ähnlich wie bei spaCy (siehe Kapitel 10.2.1), Funktionalitäten des Natural Language Processings mittels Pipelines umsetzen.

NLTK VADER

Die Python-Library NLTK stellt die Funktionalität der Gefühlsanalyse durch das regelbasierte Sentiment-Analyse-Tool **VADER** (Valence Aware Dictionary and sEntiment

Reasoner¹⁾ zur Verfügung. (SM:Web61, vgl. Singh 25.03.2021)

Dieses Tool setzt dabei auf einen Bag-of-Words-Ansatz (siehe Kapitel 9.5.2), wobei für jedes Wort aus dem Text, der analysiert werden soll, in einer Tabelle aus positiven und negativen Wörtern nachgeschlagen wird, ob es sich bei dem Wort um ein positives oder negatives Wort handelt. Dabei werden noch zusätzliche Verfahren auf das Wort angewendet. Zum Beispiel wird die Bedeutung des Wortes verschieden gewichtet, wenn es auf Wörter wie „sehr“ oder „wenig“ folgt. Außerdem wird die Bedeutung umgedreht, wenn es auf das Wort „nicht“ folgt. (SM:Web62, vgl. Terry-Jack 25.03.2021)

Der Nachteil dieses Tools ist, dass Wörter die nicht im VADER-Lexikon vorkommen, nicht beurteilt werden. (SM:Web62, vgl. Terry-Jack 25.03.2021)

TextBlob und TextBlobDE

Die Sentiment-Analyse-Funktionalitäten die von den Python-Libraries TextBlob und TextBlobDE, für jeweils englische und deutsche Texte, zur Verfügung gestellt werden, handeln nach einem vereinfachten Prinzip von NLTK VADER. Sie setzen genauso auf einen Bag-of-Words-Ansatz, verzichten jedoch auf zusätzliche Verfahren, die benachbarte Wörter bei der Beurteilung betrachten. (SM:Web62, vgl. Terry-Jack 25.03.2021)

10.3.2 Implementation

Methoden der Datenbankschnittstelle

```

1 import psycopg2
2
3 def open_cursor():
4     connection = psycopg2.connect(user="postgres",
5                                    password="postgres",
6                                    host="10.128.7.12",
7                                    port="5432",
8                                    database="AI_Boerse")
9     return connection.cursor()
10
11 def close_cursor(cursor):
12     cursor.close()
13
14 def get_Freitext():
15     cursor = open_cursor()
16     cursor.execute("""SELECT link, rohtext, datum, uhrzeit, website_url FROM
17                     freitext WHERE datum = current_date""")

```

¹⁾Zu Deutsch etwa „Wertigkeitsbewusstes Wörterbuch und Gefühlsbeurteiler“

```

17     data = cursor.fetchall()
18     close_cursor(cursor)
19     return data

```

Listing 10.7: Methoden der Datenbankschnittstelle

Abfrage der Freitexte

Zuerst werden die Freitexte aus Datenbank abgefragt und in ein Pandas DataFrame gespeichert:

```

1 data = get_Freitext()
2 df = pd.DataFrame(data = data, columns = ("link", "rohtext", "datum", "
    uhrzeit", "website_url"))

```

Aufteilung der Texte in einzelne Sätze

Mithilfe von `nltk.tokenize` werden die Texte nun in ihre einzelnen Sätze aufgeteilt. Diese Sätze werden in ein DataFrame `data_page_contents_german` gespeichert:

```

1 data_sentences = []
2 for index, row in df.iterrows():
3     for sentence in tokenize.sent_tokenize(row["rohtext"], language = "
    german"):
4         data_sentences.append([row["link"], row["datum"], row["uhrzeit"],
        row["website_url"], sentence])
5 df_sentences = pd.DataFrame(data = data_sentences, columns = ["link", "
    datum", "uhrzeit", "website_url", "sentence"])
6 german_sentences = df_sentences["sentence"].tolist()
7 data_page_contents_german = []
8 i = 0
9 for sentence in german_sentences:
10     data_page_contents_german.append([i, sentence])
11     i += 1
12 df_page_contents_german = spark.createDataFrame(data_page_contents_german).
    toDF("id", "text")

```

Listing 10.8: Aufteilung der Texte in einzelne Sätze

Übersetzung der Sätze ins Englische

Damit auch englische Sentimentmodelle zur Analyse der Sätze verwendet werden können, wird nun ein zusätzliches DataFrame `data_page_contents_english` erstellt, in dem, ins Englische übersetzte Kopien der Sätze, gespeichert werden:

```

1 translator = deep_translator.GoogleTranslator(source = "de", target = "en")
2 english_sentences = []
3 for german_sentence in german_sentences:
4     print("Translate:", german_sentence)
5     english_sentence = translator.translate(german_sentence)
6     english_sentences.append(english_sentence)
7 data_page_contents_english = []
8 i = 0
9 for sentence in english_sentences:
10    data_page_contents_english.append([i, sentence])
11    i += 1
12 df_page_contents_english = spark.createDataFrame(data_page_contents_english
13 ).toDF("id", "text")

```

Listing 10.9: Übersetzung der Sätze

Analyse der Sätze

Die Sätze werden nun mithilfe verschiedener vortrainierter Sentimentmodelle analysiert. Die Ergebnisse der verschiedenen Modelle werden in zusätzliche Spalten des DataFrames `df_sentences` gespeichert. Die Bezeichnungen der Spalten sind dabei `sentiment0`, `sentiment1`, ...

`analyze_sentiment`

```

1 pipeline = PretrainedPipeline("analyze_sentiment", lang = "en")
2 result = pipeline.transform(df_page_contents_english)
3 df_result = result.select("text", "sentiment").toPandas()
4 df_sentences["sentiment0"] = df_result["sentiment"]
5 for index, row in df_sentences.iterrows():
6     row_data = row["sentiment0"][0]
7     if row_data.result == "positive":
8         row["sentiment0"] = row_data.metadata["confidence"]
9     elif row_data.result == "negative":
10        row["sentiment0"] = str(-1 * float(row_data.metadata["confidence"]))
11    else:
12        row["sentiment0"] = 0

```

Listing 10.10: `analyze_sentiment`

TextBlob

```
1 df_result = df_page_contents_english.toPandas()
2 df_result["sentiment"] = ""
3 for index, row in df_result.iterrows():
4     df_result.loc[index, "sentiment"] = TextBlob(row["text"]).sentiment.
        polarity
5 df_sentences["sentiment1"] = df_result["sentiment"]
```

Listing 10.11: TextBlob

TextBlobDE

```
1 df_result = df_page_contents_german.toPandas()
2 df_result["sentiment"] = ""
3 for index, row in df_result.iterrows():
4     df_result.loc[index, "sentiment"] = TextBlobDE(row["text"]).sentiment.
        polarity
5 df_sentences["sentiment2"] = df_result["sentiment"]
```

Listing 10.12: TextBlobDE

NITK VADER

```
1 nltk.download('vader_lexicon')
2 sia = SentimentIntensityAnalyzer()
3 df_result = df_page_contents_german.toPandas()
4 df_result["sentiment"] = ""
5 for index, row in df_result.iterrows():
6     polarity = sia.polarity_scores(row["text"])
7     df_result.loc[index, "sentiment"] = polarity["pos"] - polarity["neg"]
8 df_sentences["sentiment3"] = df_result["sentiment"]
```

Listing 10.13: NLTK VADER

Durchschnitt der Modelle berechnen

Zu guter Letzt wird der Durchschnitt der Ergebnisse aller Sentimentmodelle für jeden Satz berechnet:

```
6           float(row["sentiment3"])) /  
7           4
```

Werte in der Datenbank speichern

Zu Beobachtungszwecken wurde eine Tabelle in der Datenbank erstellt, in der die Ergebnisse der Sentiment-Analysen gespeichert werden:

```
1 def write_sents(link, datum, uhrzeit, website_url, sentiment_eins,  
2                 sentiment_zwei, sentiment_drei, sentiment_vier, sentiment_avg):  
3     try:  
4         connection = psycopg2.connect(user="postgres",  
5                                         password="postgres",  
6                                         host="10.128.7.12",  
7                                         port="5432",  
8                                         database="AI_Boerse")  
9         cursor = connection.cursor()  
10  
11         insertQuery = """Insert Into analysierte_saetze Values (%s, %s, %s,  
12                         %s, %s, %s, %s, %s)"""  
13         recordTuple = (link, datum.strftime("%Y-%m-%d"), uhrzeit,  
14                         website_url, sentiment_eins, sentiment_zwei, sentiment_drei,  
15                         sentiment_vier, sentiment_avg)  
16  
17         cursor.execute(insertQuery, recordTuple)  
18         connection.commit()  
19     finally:  
20         cursor.close()  
21         connection.close()  
22  
23     ...  
24  
25 for index, row in df_sentences.iterrows():  
26     write_sents(row["link"],  
27                  row["datum"],  
28                  row["uhrzeit"],  
29                  row["website_url"],  
30                  row["sentiment0"],  
31                  row["sentiment1"],  
32                  row["sentiment2"],  
33                  row["sentiment3"],  
34                  row["sentimentavg"])
```

Dazu wurde das Programm täglich abends automatisiert ausgeführt.

10.3.3 Problem

Das Problem bei diesem Ansatz war, dass die Beurteilung der Sätze zu oft falschgelegen hat. Sätze, die eine Aussage zu einer aufsteigenden Prognose des DAX gemacht haben, wurden zu oft negativ beurteilt und Sätze, die über absteigende Prognosen berichtet haben, wurden zu oft positiv beurteilt.

Bewertung	Satz
-0.131488888888889	Über 14.000 Punkten dürfte ein Hochlauf bis zur Widerstandsmarke von 14.300 Punkten erfolgen.
-0.1397	Anlaufmarken nach oben bei 14.000, 14.150 und 14.300 Punkten.
0.13521230486685	Solange der DAX noch unter 13.454 Punkten bleibt, könnte sich ein tieferes Verlaufshoch herausbilden und zusammen mit dem tieferen Verlaufstief der vorherigen Abwärtsbewegung bei 13.009 Punkten den Beginn eines neuen Abwärtstrend markieren.

Tabelle 10.1: Beispiele falsch bewerteter Sätze

10.4 Analyse mithilfe eines prozeduralen Ansatzes

Die Programme, die für diesen Ansatz erstellt wurden, sind auf dem Datenträger unter X:\Programme\prozeduraler_ansatz zu finden.

Da die Ansätze mithilfe von Natural Language Processing nicht die gewünschten Ergebnisse geliefert haben, wurde sich letztendlich für einen prozeduralen Ansatz entschieden.

Dabei wurde ein Pythonprogramm geschrieben, das stündlich ausgeführt wird. Es holt sich die Texte der Börsenbriefe der letzten Stunde und zerlegt diese in ihre einzelnen Sätze. Es überprüft daraufhin ob bestimmte Stichwörter in den Sätzen vorkommen. Aufgrund dieser Stichwörter wird entschieden, ob es sich bei den numerischen Werten im Satz, um aufsteigende oder absteigende Prognosewerte handelt.

10.4.1 Verwendete Technologien

SoMaJo Tokenizer

Bei *SoMaJo* handelt es sich um einen Tokenizer, der dazu in der Lage ist, deutsche und englische Texte in ihre einzelnen Sätze zu teilen. (SM:Web60, vgl. SoMaJo Dokumentation 24.03.2021)

Da SoMaJo im Vergleich zu anderen Tokenizern, wie dem von `nltk`, bessere Ergebnisse für deutsche Texte liefert, wurde sich im Projekt AI Börse, bei der Umsetzung des prozeduralen Ansatzes, für ihn entschieden.

Zum Beispiel wird der Satz „Dies wäre z. B. ein Satz.“ vom NLTK Tokenizer zwischen „z.“ und „B.“ geteilt, während der SoMaJo Tokenizer erkennt, dass es sich dabei um einen Satz handelt.

10.4.2 Implementation

Beschaffung der Freitexte

Mithilfe der eigens geschriebenen Methode `.get_freetexte()` werden die Freitexte, die sich der Crawler innerhalb der letzten Stunde besorgt hat, in ein Pandas DataFrame gespeichert:

```
1 df_freetexte = pd.DataFrame(data = get_freetexte(), columns = ["link", "rohtext", "datum", "uhrzeit", "website_url"])
2 if df_freetexte.empty:
3     print("Keine neuen Börsennews um", datetime.now().hour, "Uhr gefunden.")
4     )
5 return 1
```

Definition der Stichwörter

Nachdem einige Börsenbriefe händisch ausgelesen und analysiert wurde, konnten Stichwortlisten aufgestellt werden, die Sätze kennzeichnen, die Aussagen über positive bzw. negative Prognosen machen. Zudem wurde eine Stichwortliste aufgestellt, die Aussagen über bereits vergangene Entwicklungen des DAX-Kurses macht.

Wochenlang wurden täglich die, vom Crawler beschafften Börsenbriefe, beobachtet und analysiert, damit die Stichwortlisten bei Bedarf erweitert werden konnten. Die Qualität

der Auswertung konnte somit verbessert werden.

```

1 up_keywords = ["oben", "Oberseite", "oberen", "oberes", "über", "Hochlauf",
    "ansteigen", "aufwärts", "steigen", "Widerstand", "Widerstände"]
2 down_keywords = ["unter", "abwärts", "Rücklauf", "unten", "Rückgang", "
    Unterstützung"]
3 curr_keywords = ["Aktuell", "konnte heute", "Widerstandsmarken", "Unterstü
    tzungsmarken", "Jänner", "Januar", "Februar", "März", "April", "Mai", "
    Juni", "Juli", "August", "September", "Oktober", "November", "Dezember",
    "Montag", "Dienstag", "Mittwoch", "Donnerstag", "Freitag", "Samstag", "
    Sonntag", "gegangen", "per Handelsschluss", "Vortag"]
```

Listing 10.14: Definition der Stichwörter

Vorbereitung der Texte

Mithilfe der Funktion `.iterrows()` wird über das DataFrame der Freitexte der letzten Stunde iteriert:

```

1 for index, row in df_freitexte.iterrows():
```

Für jeden Freitext wird der Text auf die Auswertung vorbereitet, indem zuvor bestimmte Sonderfälle behandelt werden.

Zum Beispiel soll bei Texten, in denen zwei mögliche Werte für eine Anlaufstelle (z. B. „14707/14725“) gegeben sind, nur der erste Wert („14707“) beachtet werden. Zu diesem Zweck wird der zweite Wert aus dem String entfernt.

```

1 text = row["rohtext"]
2 text = re.sub(r"([0-9]+)\. (?!Jan|Feb|Mar|Apr|May|Jun|Jul|Aug|Sep|Oct|Nov|
    Dec)", r"\1 Satzende.", text)
3 text = re.sub(r"([0-9]+)er", r"\1 er", text)
4 text = re.sub(r"(\d+\.\d+) bis (\d+\.\d+)", r"\1", text)
5 text = re.sub(r"(\d+) bis (\d+)", r"\1", text)
6 text = re.sub(r"(\d+\.\d+) (/d+\.\d+)*", r"\1", text)
7 text = re.sub(r"(\d+\.\d+) (/d+)*", r"\1", text)
8 text = re.sub(r"(\d+) (/d+)*", r"\1", text)
9 text = re.sub(r"(\d+\.\d+) ( / \d+\.\d+)*", r"\1", text)
10 text = re.sub(r"(\d+\.\d+) ( / \d+)*", r"\1", text)
11 text = re.sub(r"(\d+) ( / \d+)*", r"\1", text)
```

Listing 10.15: Vorbereitung der Texte

Aufteilung der Texte in einzelne Sätze

Mithilfe des SoMaJo-Tokenizers werden die Texte nun in ihre einzelnen Sätze aufgeteilt. Dazu muss der Tokenizer zuvor initialisiert werden:

```
1 tokenizer = SoMaJo("de_CMC")
```

Danach kann der Tokenizer verwendet werden:

```
1 sentences = tokenizer.tokenize_text([text])
2 for sentence in sentences:
3     sentence_str = " ".join([token.text for token in sentence])
```

Analyse der einzelnen Sätze

Nun wird jeder Satz mit den Stichwortlisten abgeglichen. Mithilfe der selbst geschriebenen Methode `get_dax_values_from_sentence()` werden die numerischen Werte aus dem Satz extrahiert, die realistische DAX-Werte sein könnten. Als realistisch werden Werte bewertet, die sich in einem bestimmten Bereich um den derzeitigen DAX-Wert befinden. Im Projekt AI Börse wurde sich für einen Bereich von 20% entschieden.

```
1 if contains_one_of(sentence_str, up_keywords) and not contains_one_of(
    sentence_str, down_keywords) and not contains_one_of(sentence_str,
    curr_keywords):
2     # Aufsteigende Werte erfassen
3     numbers = get_dax_values_from_sentence(sentence_str, dax)
4     up_values.extend(numbers)
5 if contains_one_of(sentence_str, down_keywords) and not contains_one_of(
    sentence_str, up_keywords) and not contains_one_of(sentence_str,
    curr_keywords):
6     # Absteigende Werte erfassen
7     numbers = get_dax_values_from_sentence(sentence_str, dax)
8     down_values.extend(numbers)
```

Speichern der Prognosewerte

Zu guter Letzt wird für jeden Freitext durch seine jeweils aufsteigenden und absteigenden Prognosewerte iteriert und die Grenze-Anlaufstelle-Paare in die Datenbank geschrieben:

```
1 # Aufsteigende Prognosen in die DB schreiben
2 for i in range(len(up_values) - 1):
3     grenze = up_values[i]
4     anlaufstelle = up_values[i + 1]
```

```
5 link = row["link"]
6 datum = datetime.now().date().strftime("%Y-%m-%d")
7 uhrzeit = datetime.now().time().strftime("%H:%M:%S")
8 write_prognose(grenze, anlaufstelle, link, datum, uhrzeit)
9
10 # Absteigende Prognosen in die DB schreiben
11 for i in range(len(down_values) - 1):
12     grenze = down_values[i]
13     anlaufstelle = down_values[i + 1]
14     link = row["link"]
15     datum = datetime.now().date().strftime("%Y-%m-%d")
16     uhrzeit = datetime.now().time().strftime("%H:%M:%S")
17     write_prognose(grenze, anlaufstelle, link, datum, uhrzeit)
```

10.4.3 Fazit

Die Ergebnisse dieses prozeduralen Ansatzes sind sehr zufriedenstellend. Es könnten jedoch Probleme entstehen, wenn man neue Quellen für Börsenbriefe hinzufügt, die ihre Texte ganz anders formulieren. Dann muss das Programm angepasst werden, sodass es auf diese neuen Formulierungen entsprechend reagieren kann.

Wie man an den verschiedenen Ansätzen sehen kann, braucht man nicht immer künstliche Intelligenz um eine Anwendung umzusetzen. Trotzdem wird man, auf lange Sicht gesehen, mit Programmen, die Natural Language Processing implementieren, bessere Ergebnisse erzielen, solange man genügend hochqualitative Trainingsdaten zur Verfügung hat.

Kapitel 11

Allgemeines über Daten

11.1 Definition

Daten ist ein weitverbreitetes Wort und wird auch gelegentlich missverstanden, da es in vielen verschiedenen Sachzusammenhängen genutzt wird. Unter jenem können zum Beispiel Angaben, Zahlenwerte, Informationen, Größen oder das Plural von Datum gemeint sein.

(AA:Web01, vgl. [duden.de](#) 10.11.2020)

Eine Erklärung zu dem Wort Daten in Verbindung zur Informatik wird in diesem kurzen Zitat erläutert:

Daten sind streng genommen einfache Informationen, die von einem Computer verarbeitet werden. Dies kann dabei auf unterschiedlichen Wegen (kopieren, bearbeiten, speichern) geschehen. Diese einfachen Informationen können dabei zunehmend komplexer werden, sodass man letztendlich unterschiedliche umfangreiche Daten antrifft. So gibt es nicht nur einfache Daten in Form eines Textes, sondern auch sehr komplexe Datenmengen wie etwa bei Musik oder Videos.

Mehrere Daten die logisch zusammengehören werden als Datensätze bezeichnet. Diese wiederum stellen Dateien gleicher Art dar.

(AA:Web02, [techfacts.de](#) 10.11.2020)

Jeden Tag werden von Milliarden von Menschen verschiedene Arten von Daten gebraucht, ohne überhaupt zu wissen, wie jene funktionieren. Egal ob MP3-Player oder DVD, viele technische Errungenschaften der Menschheit beinhalten Daten und/oder verarbeiten sie, um für uns Menschen von Nutzen zu sein und unser Leben zu erleichtern.

11.2 Wozu braucht man Daten?

In der heutigen Zeit werden haufenweise Daten gesammelt, verschickt und gespeichert. Viele Menschen nutzen und brauchen sie, um in der modernen Welt zurechtzukommen. Unsere Gesellschaft, besonders die jungen Generationen, haben sich an die Nutzung gewöhnt und könnten sich ein Leben ohne solche Daten kaum mehr vorstellen. Jeder Mensch, der einmal mit dem Internet in Verbindung kommt, sendet, speichert oder verändert Daten.

Am Beispiel Facebook zeige ich ein Szenario auf, um die Vorstellung von Daten im Internet zu erläutern. Um seinen momentanen Status auf Facebook öffentlich zu posten, muss man sich zuallererst anmelden (hierbei ist die Angabe persönlicher Daten erforderlich). Wenn dieser Prozess abgeschlossen ist hat man die Berechtigung, andere Informationen von Bekannten oder wildfremden Personen abzurufen und seine eigenen Präferenzangaben zu ändern. Eine weitere Funktion auf Facebook ist es, als Nutzer Fotos hinauf- und herunterzuladen (Speichern von Daten auf der Website oder einem lokalen Gerät).

Daten haben auch viele andere Nutzungsmöglichkeiten. Viele nutzen Daten, um sich einen Vorteil zu verschaffen und ihre Produkte leichter zu vermarkten, um somit mehr Gewinn zu erzielen. Manch anderer benutzt Informationen um sich Wissen anzueignen und sich weiter fortzubilden.

Die nächsten Exempel bieten einen kleinen Einblick in den Nutzen von Daten im Internet:

Soziale Netzwerke

Man sucht eine Plattform um Neuigkeiten und aktuelle Geschehnisse in Erfahrung zu bringen oder um neue Freunde kennen zu lernen? Dann ist man bei sozialen Netzwerken am rechten Platz. Auf sozialen Netzwerken findet man von Katzen-Videos bis hin zu Aufnahmen eines berühmten Sängers so gut wie alles. Plattformen wie Facebook, Instagram und Whatsapp haben einen großen Einfluss auf die moderne Gesellschaft und sind daher eine immense Geldquelle für große Unternehmen. Solche Plattformen können Informationen eines Nutzers analysieren und zum Beispiel das nächste Schnellimbiss Restaurant für den Standort eines Benutzers anzeigen lassen. Deshalb sollte man sich als Nutzer ausreichend mit dem Thema Privatsphäre im Internet und den Sicherheitseinstellungen des Webbrowsers auseinandersetzen.

Werbung und Cookies

Es gibt sie überall, egal ob auf einem Flyer, auf Plakaten oder im Internet. Die Rede ist von Werbung. Sie wird benutzt, um einem Unternehmen mehr Kundschaft zu verschaffen. Je mehr Leute die Werbung eines Unternehmens sehen, desto wahrscheinlicher ist es, dass sie diese als Kunden gewinnen. Was ist dafür besser geeignet als das World Wide Web?

Man kennt das Phänomen: Man sucht auf einer Shopping-Webseite nach bestimmten Waren und in darauffolgender Zeit werden Anzeigen von derselben Ware auf anderen Seiten angezeigt. Dieses Ereignis ist leicht erklärt. Durch Surfen im Internet werden Spuren hinterlassen. Dies können Informationen darüber sein welche Seiten besucht, was dort angeklickt wurde und welches Gerät benutzt wurde. Solche Daten werden über sogenannte Cookies¹ gesammelt. Cookies haben auch ihre positiven Seiten, wie zum Beispiel das Speichern der Daten im Warenkorb einer Shopping-Webseite. Viele Webseiten würden ohne Cookies sehr schlecht bis gar nicht funktionieren.

Suchmaschinen

Das Geschäftsmodell von Suchmaschinen wie Bing, Google und Co. unterscheidet sich wenig von jenem der Shopping-Webseiten. Suchmaschinen speichern die Suchanfragen der Nutzer und analysieren ihr Verhalten. Dies wird genutzt, um bei folgenden Suchanfragen ähnliche Ergebnisse zu liefern, die dem Nutzer gefallen könnten. Solche Daten können aber auch genutzt werden, um auf jeden Benutzer genau abgestimmte Werbung anzeigen zu lassen.

Wie wir an vorherigen Beispielen gesehen haben, sind Daten für viele Zwecke zu gebrauchen. Auch wenn ein Großteil eher negativ auffällt, sollte man jedoch berücksichtigen, dass wir Daten tagtäglich benutzen und sie unseren Alltag erleichtern. Solange man im Gewissen festhält, was mit seinen Daten passiert, muss man sich beim Surfen im Internet keine großen Sorgen machen.

(AA:Web03, vgl. Anita Vetter 4.11.2020)

¹Textdateien, die Webseiten bei einem Besuch automatisch auf dem Rechner abspeichern

11.3 Big Data

Der Begriff Big Data wird wohl fast jedem bekannt sein, andernfalls wird es hier nochmals erklärt:

Der Begriff Big Data stammt aus dem englischen Sprachraum. Erst als Phänomen oder als Hype wahrgenommen, fassen die Experten mittlerweile unter diesem Begriff zwei Aspekte zusammen. Demnach umschreibt er zum einen die immer rasanter wachsenden Datenmengen; zum anderen aber geht es auch um neue und explizit leistungsstarke IT-Lösungen und Systeme, mit denen Unternehmen die Informationsflut vorteilhaft verarbeiten können - Stichwort Machine Learning. Insbesondere unstrukturierte Daten - zum Beispiel aus den sozialen Netzwerken - machen dabei einen nicht unerheblichen Teil der Massendaten aus.

(AA:Web04, Michael Radtke 04.12.2020)

Bei Big Data geht es nicht um die Anzahl der Daten sondern darum, was Sie damit machen. Big Data hat einen sehr großen Nutzen und eine große Auswahl an Quellen. Es könnte der Kostensenkung, Zeitsparung, Produktentwicklung oder cleveren Geschäftentscheidungen dienen. In Kombination mit Data Analytics kann man die verschiedensten Aufgaben einfacher bewältigen.

(AA:Web05, vgl. sas.com 04.12.2020)

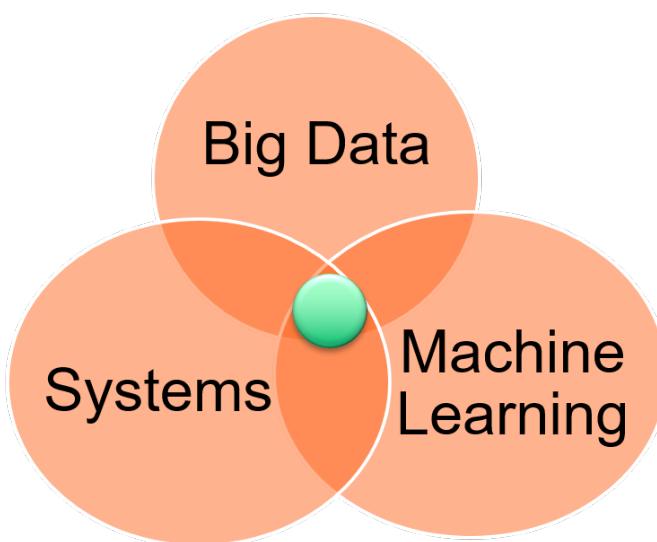


Abbildung 11.1: Schnittpunkt von BigData / ML und Systems
Quelle: (AA:Web23, vgl. mycourses.aalto.fi 25.02.2021)

Hier werden Aufgaben und dazu passende Beispiele aufgelistet, mit denen Sie sich einen kleinen Einblick in die Welt von Big Data verschaffen können:

Sofortige Problemlösung im Unternehmen

Als Beispiel: Die Autoindustrie setzt seit neuestem auf Big Data und Data Analytics, diese zwei Methoden helfen dem Unternehmen, Fehler und Probleme nahezu in Echtzeit zu finden und zu lösen. Informationen der Geräte werden regelmäßig hin und her gesendet und sollte mal eine Maschine einen Defekt haben, so wird dieser auch in den Daten erkennbar sein. Somit können die Fehlfunktionen der Geräte präzise und ohne Hilfe eines Menschen behoben werden.

In erster Linie wird Big Data gebraucht, um bessere Entscheidungsgrundlagen für Geschäftstätigkeiten zu schaffen. Als Grundlage dafür kann man die verschiedensten Quellen auswählen, wie zum Beispiel Berichte von sozialen Medien, Informationen von Sensoren, Einzelverbindnungs nachweise von Mobiltelefonen oder Internet-Clickstreams.

Wettbewerbsvorteil mittels Big Data Analytics

Besonders Unternehmen nutzen Big Data zu ihrem Vorteil. Durch steigende technologische Entwicklung erhöhen sich die Datenmassen. Diese werden von Firmen analysiert. Die Ergebnisse dieser Analyse bieten ganz neue Einblicke in die Interessen, das Kaufverhalten und auch das Risikopotenzial von Kunden sowie potenziellen Interessenten.

Damit die Daten auch entsprechend gefiltert, untersucht, beurteilt und passend eingruppiert werden können, greifen Unternehmen gezielt zu Analytics-Methoden. Diese Erkenntnisse können dem Unternehmen einen Wettbewerbsvorteil gegenüber ihren Konkurrenten bringen, indem sie mit den gesammelten Informationen effektiveres Marketing betreiben und dadurch zu mehr Umsatz kommen.

Software Tools für eine fortschrittliche Analytik

Unternehmen verfolgen das Ziel, eine bessere Entscheidungsgrundlage für die eigene Geschäftstätigkeit zu konzipieren. Dies wird durch komplexe Datenanalyse möglich, die von Experten für Big Data ausgewertet werden. Die sogenannten „Data Scientists“ analysieren große Mengen von entsprechenden Transaktionsdaten und anderweitigen Informationen aus den unterschiedlichsten Datenquellen, um damit das Ziel des Unternehmens zu erreichen.

Für die Verarbeitung und Analyse dieser Massendaten greifen Unternehmen auf Software Tools zurück, die Big Data Analytics umfassend ermöglichen.

Optimierte Speichertechnik

Big Data spielt nicht nur bei hochmoderner Software, sondern auch in vielen Gebieten wie der Hardware eine große Rolle. Selbst bei der Speichertechnologie hat Big Data eine entscheidende Bedeutung. Mittlerweile macht es die Speichertechnologie möglich, Datenvolumen im Rahmen des sogenannten In-Memory Computing direkt im Hauptspeicher eines Rechners zu halten. Bevor dies eine Möglichkeit war, mussten Daten gewöhnlich auf langsamere Speichermedien wie HDDs oder Datenbanken ausgelagert werden. Nun wird dank In-Memory Computing sowohl die Rechengeschwindigkeit deutlich erhöht als auch die Echtzeitanalyse umfangreicher Datenbestände ermöglicht. (AA:Web06, vgl. Michael Radtke 21.12.2020)

Dies ist jedoch nur ein kleiner Teil vom großen Kuchen. Big Data wird von Tag zu Tag weiterentwickelt und man findet jedes Mal eine neue Nutzungsmöglichkeit. Wir alle sollten gespannt sein, was die Zukunft mit der riesigen Flut von Daten mit sich bringt.

11.3.1 Data Analytics

Wer sich schon einmal mit Big Data auseinander gesetzt hat, ist sehr wahrscheinlich schon auf das Wort Data Analytics gestoßen, welches im deutschen Sprachgebrauch Datenanalyse genannt wird. Data Analytics ist die Wissenschaft der Analyse von Rohdaten, um Schlussfolgerungen aus diesen Daten zu ziehen. Die meisten Techniken und Prozesse der Data Analytics wurden zu mechanischen Prozessen und Algorithmen automatisiert.

Man kann Datenanalyse für verschiedene Zwecke nutzen. Die geläufigste Anwendung ist das Aufdecken von Trends und Metriken, die sonst in der Informationsmenge verloren gehen würden. Diese Informationen können anschließend zur Optimierung von Prozessen verwendet werden, um die Gesamteffizienz eines Unternehmens zu steigern.

Ein Vorteil von Data Analytics ist es, dass jede Art von Information den Datenanalysetechniken unterzogen werden kann, um Einblicke zu erhalten oder eine Verbesserung von bestimmten Methoden zu ermöglichen.
(AA:Web07, vgl. Jake Frankenfield 26.12.2020)

Der Prozess von Data Analytics umfasst verschiedene Schritte:

1. Bestimmung von Datenanforderungen oder Gruppierung der Daten; Daten können nach Alter, Wohnort, Geschlecht und vielen anderen Kriterien getrennt werden
2. Prozess der Erfassung, kann über eine Vielzahl von Quellen wie Internet, Bücher, Personenbefragungen, Statistiken und Zeitungen erfolgen
3. Organisation der Daten zur Analyse; mithilfe verschiedener Formen von Software, die statistische Daten aufnehmen kann, möglich
4. Bereinigung nach abgeschlossener Analyse; Überprüfung zur Sicherstellung, dass keine redundanten Daten aufgenommen werden

Data Analytics ist wichtig da sie Unternehmen dabei hilft, ihre Leistung zu optimieren und mögliche Trends vorherzusagen. Durch die Implementierung in das Geschäftsmodell können Unternehmen zur Kostensenkung beitragen, indem sie große Datenmengen speichern und praktischere, beziehungsweise bessere Geschäftsmethoden ermitteln.

The Process

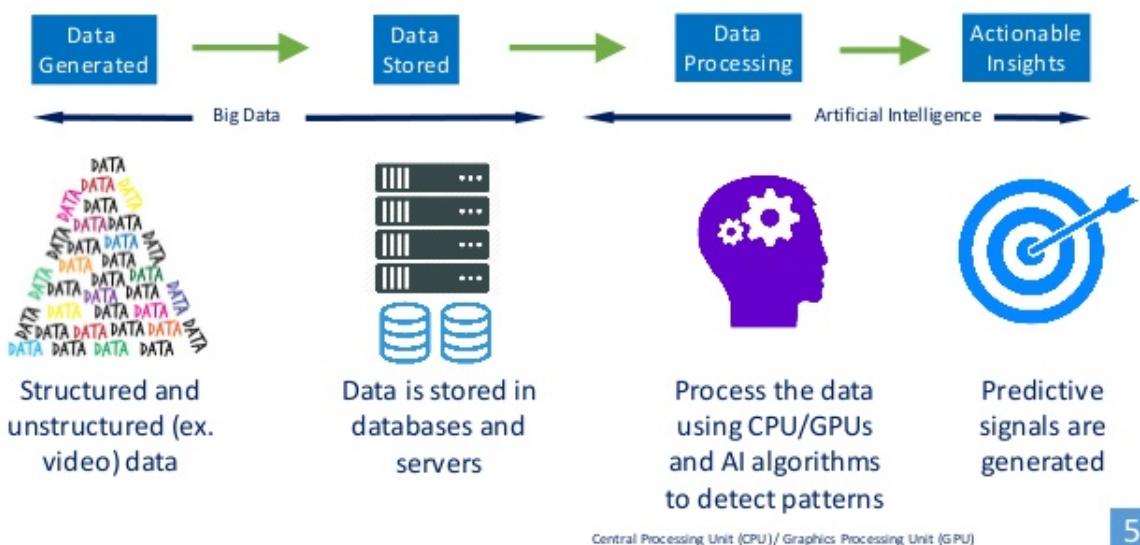


Abbildung 11.2: Ablauf der Daten
Quelle: (AA:Web16, vgl. devopedia.org 01.02.2021)

11.3.2 Predictive Selling

Wie schon im vorigen Thema erwähnt gibt es mehrere Möglichkeiten, Datenanalyse zu benutzen. Eine davon ist Predictive Analytics. Predictive Analytics befasst sich mit der Analyse von Daten und zeigt wahrscheinlich passierende, zukünftige Ereignisse. Die Daten, die zur Prognose genutzt werden, sind zum Beispiel Verkaufszahlen des letzten Sommers. Data Analytics Experten stellen sich Fragen wie „Welche Marken waren begehrt oder welche Kleidung wurde häufig gekauft?“. Durch die Information, welche Produkte gekauft wurden, kann man Prognosen treffen und effizienter investieren.

Als Beispiel ein Online-Shop: Dieser kann die Käufe seiner Kunden analysieren und den einzelnen Kunden die durch die Prognosen bestimmten individuellen Produktgruppen vorschlagen und erzielt dadurch möglicherweise mehr Profit. In Extremfällen werden dem Kunden sogar die prognostizierten Produkte zugeschickt der dann entscheiden kann, ob er das möglicherweise sogar verbilligte Produkt behält oder zurückschickt. Solch eine Art von Verkauf wird in Zukunft sehr wahrscheinlich häufiger von modernen Unternehmen genutzt werden, da diese sehr gewinnbringend scheinen.

11.4 Künstliche Intelligenz

11.4.1 Definition

Künstliche Intelligenz (KI), oder auch Artificial Intelligence (AI) genannt, simuliert die menschliche Intelligenz mithilfe von Maschinen, insbesondere Computersystemen. Dies beinhaltet die Erfassung von Informationen und Regeln für Verwendung der Informationen, die Verwendung der Regel, um Näherungen oder genaue Schlussfolgerungen zu prognostizieren und die Selbstkorrektur.

(AA:Web08, vgl. ComputerWeekly.de 29.12.2020)

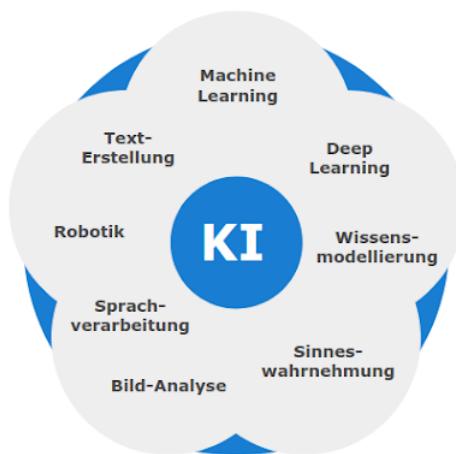


Abbildung 11.3: Themengebiete die KI benutzen
Quelle: (AA:Web17, vgl. Mathias Sauermann 01.02.2021)

11.4.2 Die Geschichte hinter der künstlichen Intelligenz

Der erste Grundstein für das, was wir heute unter künstlicher Intelligenz verstehen, wurde schon im Jahre 1936 vom britischen Mathematiker Alan Turing gelegt. Turing bewies durch seine Theorien, dass eine Rechenmaschine (auch „Turingmaschine“ genannt) in der Lage wäre, kognitive² Prozesse auszuführen, sofern diese sich in mehrere Einzelschritte zerlegen und durch einen Algorithmus darstellen lassen.

Im Sommer 1956 trafen sich Wissenschaftler zu einer Konferenz am Dartmouth College

²Der Begriff kognitiv bezeichnet Funktionen des Menschen, die mit Wahrnehmung, Lernen, Erinnern, Denken und Wissen in Zusammenhang stehen.

im US-Bundestaat New Hampshire, um über die Aspekte des Lernens von Maschinen zu diskutieren. John McCarthy, ein Programmierer, hält den Begriff „künstliche Intelligenz“ für passend und schlägt diesen auch vor. Im selben Jahr wird das erste KI-Programm mit dem Namen „the first artificial intelligence program“ fertig gestellt, welches mehrere Dutzend mathematische Lehrsätze beweisen kann.
(AA:Web09, vgl. bosch.com 29.12.2020)

11.4.3 Arten von KI

Durch den technischen Fortschritt der letzten Jahrzehnte kam es zum Ausschwung beim Thema künstliche Intelligenz. Neue Technologien eröffnen neue Möglichkeiten wie maschinelles Lernen, neuronale Netzwerke, natürliche Sprachverarbeitung, genetische Algorithmen und rechnerische Kreativität.

Die KI entwickelt sich mit jedem Tag immer weiter. Man unterscheidet:

- **Rein reaktive künstliche Intelligenz**

Dieser Typ der künstlichen Intelligenz ist die grundlegenste Form der aufgelisteten Typen. Dieses System hat keine Vorstellung von der äußeren Welt. Es erkennt nur die Situation, in der es sich gerade befindet und reagiert dementsprechend darauf. Diese Form von KI bildet keine Erinnerungen und kann auch nicht auf vergangene Erfahrungen zurückgreifen, um aktuelle Entscheidungen zu beeinflussen. Rein reaktive KI wird benutzt, wenn es sich auf einen Bereich spezialisieren soll.

Beispiele: IBMs Deep Blue, das Kasparov beim Schach geschlagen hat, Googles AlphaGo, das über den menschlichen Champion beim Go-Spiel triumphierte

- **Systeme mit begrenztem Gedächtnis**

Ein weiterentwickelter Typ der künstlichen Intelligenz ist das System mit begrenztem Gedächtnis. Dieses System speichert Informationen von Teilen der Vergangenheit und benützt diese um aktuelle Situationen besser zu bewältigen. Das System hat gerade noch genug Gedächtnis oder Erfahrungsvorrat, um richtige Entscheidungen zu treffen und entsprechende Handlungen auszuführen.

Beispiele: selbstfahrende Fahrzeuge, Chatbots und persönliche digitale Assistenten.

- **Systeme mit eigenem Bewusstsein**

Theory of Mind bedeutet, ein eigenes Bewusstsein zu entwickeln. Zu Theory of Mind gehört, sich und andere erkennen zu können, aber auch ein Verständnis

dafür zu entwickeln, dass andere sich irren und deswegen auch falsch handeln könnten.

Solch eine Art von KI hat die Fähigkeit, Gedanken und Emotionen zu verstehen, die das menschliche Verhalten beeinflussen. Maschinen die solch ein System hätten, könnten ein Verständnis von Gefühlen, Motiven, Absichten und Erwartungen sowie die Fähigkeit darauf aufbauend sozial zu interagieren erlernen und nützen. Zur aktuellen Zeit gibt es noch kein solches System, jedoch sind solche Systeme wahrscheinlich die nächste Stufe der KI.

Beispiele aus dem Science Fiction: C-3PO und R2-D2 aus den Star Wars Filmen, sowie Sonny aus dem Film „I Robot“

- **Sich „ihrer selbst“ bewusste Systeme**

Diese Form von KI kann Vorstellungen über sich selbst bilden. Sie stellen eine Erweiterung der Systeme mit eigenem Bewusstsein dar. Dieses System kann eigene Schlussfolgerungen ziehen, Gefühle anderer vorhersagen und Abstraktionen bilden. Sie sind sich ihrer inneren Zustände bewusst und sind die zukünftige Maschinengeneration, dazu gehört super intelligent, empfindungsfähig und bewusst.

Beispiele: Eva aus dem Film „Ex Machina“ und Synths aus der TV-Serie „Menschen“

(AA:Web10, vgl. Dr. Hansjörg Leichenring 30.12.2020)

11.5 Verarbeitung von Daten

11.5.1 Was ist CRUD?

Popularität erlangte CRUD im Jahr 1983. In diesem Jahr veröffentlichte der britische IT-Berater und Autor James Martin sein Buch „Managing the DataBase Environment“ und prägte darin den Begriff. CRUD ist ein Wort aus der Welt der Informatik und steht für Create, Read, Update und Delete. Das sind die grundlegenden Operationen, die zum Beispiel in einer Datenbank ausgeführt werden können.

Die vier Operationen haben verschiedene, aber zusammenhängende Zwecke:

- Create ist eine Anweisung, bei der ein neues Datenobjekt erstellt wird
- Read liest Daten aus einer Tabelle basierend auf den Eingabeparametern aus
- Update verändert bereits bestehende Datenobjekte und überschreibt diese mit den neuen Daten
- Delete löscht Daten aus einer angegebenen Tabelle

Mit diesen Funktionen lassen sich Daten in Systemen anlegen und wie in der Aufzählung schon beschrieben, entsprechend verwalten.



Abbildung 11.4: Darstellung von CRUD
Quelle: (AA:Web18, vgl. dev.to 01.02.2021)

Verwendungszwecke von CRUD

Prozesse die auf CRUD basieren, kommen sowohl bei der Programmierung von Applikationen, als auch bei der Verwaltung von Datenbanken zum Einsatz.

Geübte Datenbankadministratoren benutzen CRUD-Operationen beispielsweise, um Probleme in Datenbanken zu überprüfen und sie zu lösen oder umfangreiche Bereinigungen durchzuführen. Der Endanwender benutzt CRUD beispielsweise, um einen Account anzulegen, ihn zu nutzen, ihn zu verändern oder bei Bedarf zu löschen.

Aufgrund der grundlegenden Operationen kann CRUD für Programmierer sehr nützlich sein. Deswegen sollten Anwendungen in der Lage sein, alle Aktionen mit den vier CRUD-Operationen ausführen zu können. Sollte eine Aktion nicht mit den CRUD-Operationen ausführbar sein, sollte ein eigenes Modell geschaffen werden.

Sprachumgebungen die CRUD unterstützen

CRUD Funktionen werden in verschiedenen Plattformen und Sprachumgebungen eingesetzt. Beispiele für Sprachumgebungen, die CRUD-Operationen unterstützen:

- Java
- JavaScript
- PHP
- .NET
- Perl
- Python

Die Ausführung von Create, Read, Update und Delete ist von Programmiersprache zu Programmiersprache anders. In SQL, das sehr bekannt ist Relationale Datenbanken zu verwalten, heißen die vier Befehle Insert (einfügen), Select (lesen), Update (aktualisieren) und Delete (löschen).

Bei RESTful HTTP, das meistens für die Umsetzung von Webservices verwendet wird, heißen die vier Befehle hingegen POST (anlegen), GET (lesen), PUT (anlegen oder aktualisieren) und DELETE (löschen).

Nachteile von CRUD

Die zuvor genannten Operationen sind oftmals nicht differenziert genug, um alle Anforderungen von modernen Applikationen zu erfüllen.

Bei näherer Betrachtung ergeben sich mehrere Fragen, die in diesem Fall geklärt gehören. Beispielsweise sollte geklärt werden, ob ein Unterschied zwischen dem Entfernen von bereits erledigten und unerledigten Aufgaben besteht. Offene Aufgaben könnten etwa gelöscht werden, weil sie hinfällig sind. Hingegen werden erledigte Aufgaben entfernt, um Ordnung in Listen zu bringen. Die Einträge können weiterhin von Interesse sein, um im Nachhinein beispielsweise eine Übersicht der erledigten Aufgaben darzustellen.

Weit vorausdenkend lässt sich der Inhalt einer Aufgabe ändern. Ebenso wie die erledigten Einträge, ist dies ein Aktualisierungsvorgang. Jedoch unterscheiden sich die zwei Operationen stark voneinander.

Noch ein großer Makel ist, dass CRUD im Standard keine Historie liefert. Vielmehr führen die Aktualisierungs- und Löschoperationen zu einer nicht mehr veränderbaren Zerstörung von Daten oder ganzen Datensätzen. Finden also beispielsweise mehrere Änderungen an einem Objekt statt, lassen sich die einzelnen Schritte der Bearbeitung somit nicht mehr nachvollziehen. Dieses Problem lässt sich nur umständlich, durch mehrere zusätzliche Felder oder durch eine neue Tabelle lösen, in der alte Bearbeitungszustände abgelegt werden.

(AA:Web11, vgl. mindsquare.de 31.12.2020)

11.5.2 Unterscheidung der Daten

Es gibt viele Möglichkeiten, Daten zu klassifizieren. Eine solche ist es, Daten nach ihrem Aufbau (physische Sicht) und ihrem Inhalt (logische Sicht) zu unterscheiden. Wenn Daten in eine Dateneinheit zusammengefasst werden können, heißen diese formatierte Daten. Daten für betriebliche Anwendungen sind meist formatierte Daten, die von einem Anwendungsprogramm unter einem bestimmten Namen ansprechbar sind und deren Länge bekannt ist.

(AA:Web12, vgl. finanzen.net 01.01.2021)

Der Gegensatz zu formatierten Daten sind unformatierte Daten. Unformatierte Daten, sind Daten wie durchgehende Texte, Daten, die in Form einer Datei gespeichert sind und nicht weiter strukturierte Daten. Unformatierte Daten werden auch als unstrukturierte Daten bezeichnet und sind ungeeignet für maschinelle Interpretationen, daher werden sie meist zu formatierten Daten verarbeitet.

Daten die auf unbestimmte Zeit gespeichert werden nennt man Stammdaten und sind in relationalen Datenbanken sehr geläufig. Kundendaten einer Firma sind beispielsweise Stammdaten und beinhalten Attribute wie Kundennamen, Kundennummer, Telefonnummer, Gehalt oder Adresse.

Bewegungsdaten werden jene Daten genannt, die nur für einen absehbaren Zeitraum angelegt sind. Geleistete Arbeitsstunden pro Monat sind beispielsweise Bewegungsdaten, die für einen kurzen Zeitraum angelegt werden.

(AA:Web13, vgl. wirtschaftslexikon24.com 01.01.2021)

11.5.3 Interpretation und Analyse von Kursverläufen

Was ist die Börse?

Bevor ich erkläre, was eine Börse im Zusammenhang mit Big Data, Informatik oder Technik Allgemein zu tun hat, sollte man wissen, was eine Börse überhaupt ist und wie diese funktioniert. Eine gute und kompakte Beschreibung liefert dieses Zitat:

Die Börse ist ein Marktplatz an dem Waren und Güter, wie Wertpapiere, Rohstoffe, Devisen oder Derivate, in einem regulierten Umfeld gehandelt werden. Im Englischen spricht man von der Stock Exchange.

Bei einer Börse treffen Käufer und Verkäufer aufeinander. Der Preis eines gehandelten Gutes wird durch Angebot und Nachfrage bestimmt. Ist die Nachfrage hoch, steigt der Preis. Ist das Angebot hoch, sinkt der Preis. Der Börsenkurs ist das Abbild vergangener Preise.

(AA:Web15, ig.com 06.01.2021)

Börsen-Prognosen mittels Big Data Analytics

Natürlich hat die Informatik auch die Zukunft der Börsenwelt drastisch verändert. Das Potenzial, dass die Massen an Daten bringen kann genutzt werden, um zum Beispiel den Börsenkurs vorherzusagen.

Technologieunternehmen versuchen mittels Social Media Netzwerken wie beispielsweise Twitter richtige Prognosen zu treffen. In diesem Fall wird anhand von Twitter-Nachrichten mithilfe einer Sentimentanalyse die Stimmung der Twitter-Nutzer mit den Veränderungen des Börsenkurses verglichen. Amazon und Facebook sind sehr weit fortgeschritten, wenn es um dieses Thema geht und übernehmen solche Analysen.

Auf momentanen Stand, zeigen die Twitter-Sentiments von beiden Unternehmen eine positive Korrelation an. Um zu den ausgewerteten Prognosen zu kommen, benötigt man mehrere Schritte:

1. Untersuchen, ob eine Beziehung zwischen den Twitter-Sentiments und dem Börsenkurs besteht
2. Überprüfen, ob die Twitter-Sentiments eine Voraussagekraft für die Veränderung des Börsenkurses haben

(AA:Web14, Digitale Transformation und Unternehmensführung 06.01.2021)

Vergleich im Projekt AI Börse

Die Projektgruppe „AI Börse“ beschäftigt sich mit dem auswerten der Börsen-Prognosen. Das Ziel des Projektes ist es darzustellen, welche Quelle den Börsenverlauf über einen bestimmten Zeitraum am besten anzeigt.



Abbildung 11.5: Logo AI Börse
Quelle: vgl. Projektgruppe AI Börse 01.01.2021

Die Angehensweise zum Prognostizieren von Börsenkursen und das Auswerten von Börsen-Prognosen haben viele Ähnlichkeiten miteinander:

Datenbeschaffung

Es ist sehr wichtig glaubwürdige Quellen zu besitzen, denn ohne richtiger Dateneingabe gibt es auch keine richtigen Prognosen oder Auswertungen. Die Auswahl der Quellen muss sehr sorgfältig geprüft und ständig kontrolliert werden. Jedoch ist nicht nur die Auswahl der Quellen ein Problem, sondern auch die Beschaffung der Daten aus den Quellen. Im Normalfall sind die Daten unformatiert und somit schwer zu verwerten. Es gibt mehrere Methoden die unformatierten Daten zu verarbeiten, jedoch geschieht das in dem Projektteam durch einen sogenannten „Crawler“³. Der Crawler durchsucht die angegebenen Webseiten und schickt die gefilterten Daten in die Datenbank.



Abbildung 11.6: Verarbeitung von Daten
Quelle: (AA:Web19, vgl. thegrammarlab.com 01.02.2021)

Techniken

Um Börsenkurverläufe zu prognostizieren muss man die großen Datenmenge auswerten. Dies kann durch verschiedene Methoden wie zum Beispiel eine Sentimentanalyse geschehen, jedoch ist das nicht vonnöten, wenn es um das Auswerten von Börsen Prognosen geht. Das Vergleichen von der heutigen Prognose mit dem morgigen Börsenstand kann sogar händisch abgearbeitet werden, jedoch ist dies auf langfristiger Sicht nicht sehr praktisch. Es gibt verschiedene Möglichkeiten das Auswerten zu automatisieren, eine davon ist ein Python Programm zu schreiben das den heutigen Stand der Börse mit den vergangenen Prognosen abgleicht und einschätzt.

Genauigkeit der Ergebnisse

Um eine hohe Genauigkeit zu erzielen braucht man sehr viele Daten. In diesem Fall zählt das Motto „Je mehr desto besser!“. Besonders eine künstliche Intelligenz profitiert davon, weil diese aus jeder weiteren Information mehr lernen und ihre Antworten darauf

³Computerprogramm, das automatisch Dokumente im Web durchsucht

abstimmen kann. Im Fall Prognosen aufzustellen, bräuchte man viele Daten aus der Vergangenheit, um die Zukunft möglichst gut vorherzusagen. Im Vergleich dazu braucht man zum Analysieren von Börsennews weniger Daten von der Vergangenheit. Man sollte, um eine hohe Genauigkeit zu erzielen, Daten über einen Zeitraum sammeln und sie jeden Tag abgleichen.

Unterschiede der Nutzung des Produkts

Die beiden Tätigkeiten haben verschiedene Nutzen, obwohl sie so viele Ähnlichkeiten bei der Gewinnung der Information haben. Das Ziel von der Prognose des Börsenkurses ist es, eine möglichst genaue Vorhersage zu treffen, meist mittels künstlicher Intelligenz. Börsen-Prognosen helfen Anlegern, ihr Vermögen möglichst gut anzulegen, jedoch muss sie nicht immer stimmen. Im Gegensatz zu der Prognostizierung des Börsenkurses ist das Ziel für die Auswertung von Börsennews ein anderer. Das Ergebnis soll den Anlegern zeigen welche Quellen glaubwürdiger sind und welche eher dazu neigen falsch zu liegen.

KI steuert Businessentscheidungen

Die künstliche Intelligenz ist schon etwas Erstaunliches. Sie bietet den Menschen eine neue Dimension von Möglichkeiten, womit wir unserer Gesellschaft eine große Stütze bieten können. Künstliche Intelligenz kann so gut wie überall genutzt werden, egal ob in Fabriken oder Supermärkten. KI ist überall zu gebrauchen. Ganz besonders Großunternehmen bereichern sich durch die Nutzung von künstlicher Intelligenz, da es sehr viele Arten und Wege gibt, diese dort zu benutzen.

Durch die Nutzung von verschiedenen Analysemethoden und künstlicher Intelligenz kann man die Prozesse automatisieren, Verkaufsfläche optimieren, die Kundenzufriedenheit steigern und höhere Erträge erzielen. Hier werden ein paar Methoden vorgeführt, wie KI und Analysemethoden eingesetzt werden, um dem Handel zu helfen:

Path Analytics

Die Möglichkeit die sich durch Path Analytics ergibt ist, dass zum Beispiel Kundenverhalten und die Kundeninteraktionen zwischen Online und Offline genauer nachvollziehbar sind. Mittels Sankey-Diagramm können auch die Besucherverhalten grafisch auf einer Website dargestellt werden.

Die meistbesuchten Pfade und wie die Kunden auf der Website navigieren können, wird dadurch analysiert und zum Vorteil für das Unternehmen genutzt. Path Analytics

kann Antworten auf die Frage liefern, welche Pfade zu einem Kauf führen und welche nicht. Das Zusammenführen, Analysieren und Darstellen der Daten wird als Makrosicht dem Einzelhändler zur Verfügung gestellt, somit hat er eine Übersicht wie sich das Kundenverhalten auswirkt.

Dwell Analytics

Durch die Methode Dwell Analytics kann ermittelt werden, wie viel Zeit Kunden an einem bestimmten Ort verbringen. Lange Wartezeiten könnten auf schlechte Kundenbetreuung hinweisen oder darauf, dass gesuchte Produkte nicht so leicht zu finden sind. Diese Methode kann praktisch zum Beispiel mit einer Sicherheitskamera geschehen, diese beobachtet wie Objekte, in diesem Fall Kunden, auf dem vordefinierten Bereich verweilen.

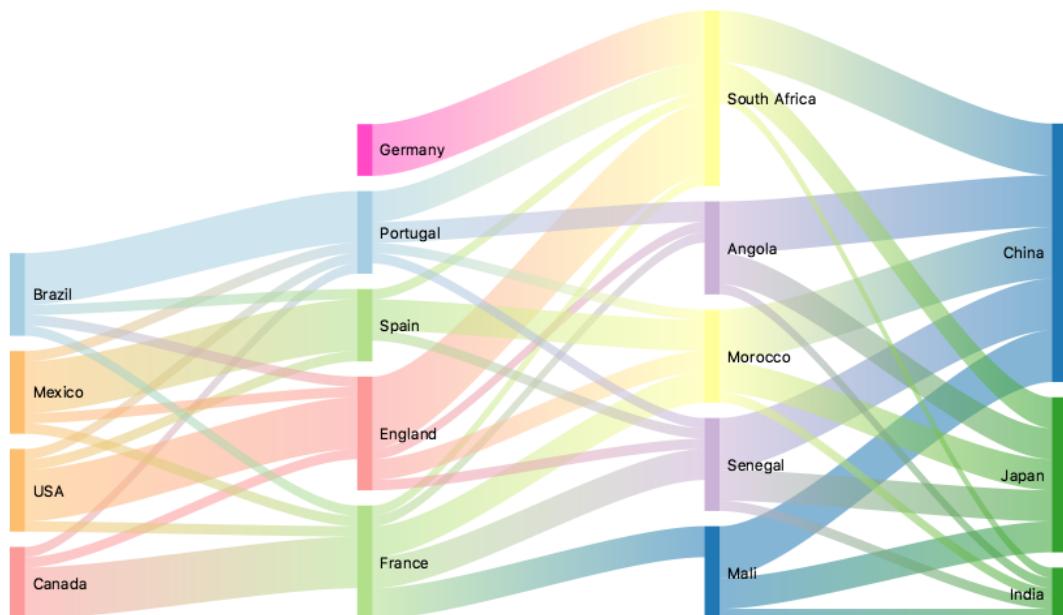


Abbildung 11.7: Beispiel für ein Sankey-Diagramm
Quelle: (AA:Web24, vgl. harmoniccode.blogspot.com 25.02.2021)

Sankey Diagramm

Diese Diagramme sind praktisch, um den Datenfluss zwischen mehreren Elementen auf verschiedenen Ebenen und die Datenzusammensetzung eines oder mehrere Elemente zu veranschaulichen.

Sankey-Diagramme veranschaulichen Knoten und Verknüpfungen. Die Breite der Verbindungen zwischen den Knoten ist proportional zur Größe der Flussmenge.
(AA:Web25, vgl. Mustapha Mekhatria 27.02.2021)

Queue Analytics

Die Queue Analytics ermitteln wie lange sich ein Kunde in der Warteschlange befindet. Diese Zeit kann darauf hinweisen, dass die Kunden schlecht oder zu lange betreut werden oder lange beim Einkaufen warten müssen.

Sentiment Analytics

Sentiment Analytics kann gut mit den anderen Methoden umgesetzt werden. Die Methode kann die Stimmung des Kunden erkennen, indem die Gesichter der Kunden analysiert werden und dementsprechend darauf reagiert werden kann. Kleinigkeiten wie zum Beispiel Frustration beim Warten in der Warteschlange zur Kassa oder Freude über ein Sonderangebot können durch die Sentiment Analytics herausgefunden werden.

Die Sentiment Analytics kann nicht nur durch Bilder und Aufnahmen die Stimmung der Kunden analysieren, in anderen Branchen wie zum Beispiel Börsen können Texte analysiert werden und man kann theoretisch ablesen, ob laut einer Börsenprognose der Kurs fällt oder steigt.

Die Sentiment Analyse auf Börsenprognosen zu verwenden war der Ansatz für das Projektteam AI Börse, da es sehr schwer zu lesende Prognosen gibt. Diese Methode war sehr vielversprechend, jedoch hat sie schlechte Ergebnisse geliefert und dieser Ansatz wurde schlussendlich verworfen.

In-Store-Engagement

Die In-Store-Engagement Methode erkennt, wenn sich Kunden mit einem Produkt, einem Verkäufer oder einer Beschilderung auseinandersetzen. Durch diese Analyse kann man Ergebnisse bekommen, die zeigen wie viel Zeit ein Kunde mit einem Produkt in der Hand verbracht hat oder im Gespräch mit einem Verkäufer war.
(AA:Web26, vgl. Thomas Willems 27.02.2021)

Kapitel 12

Repräsentation von unformatierten Daten

12.1 Strukturen von Daten

12.1.1 Was sind unformatierte Daten?

Informationen die in einer nicht identifizierbaren und nicht normalisierten Datenstruktur vorliegen, werden unformatierte Daten genannt. Unter diesen Daten kann man sich beispielsweise Textdateien, Präsentationen, Videos, Audiodaten, aufgezeichnete Sprache oder Bilder vorstellen, da diese nicht strukturiert sind. Besonders im Big Data Umfeld haben unformatierte Daten eine große Bedeutung, da sich viele relevante Informationen darin befinden.

12.1.2 Struktur der Daten

Es gibt Unterschiede, wenn es um das Thema digitale Daten geht. Bei diesen wird zwischen unstrukturierten und strukturierten Daten unterschieden. Der große Unterschied ist, dass strukturierte Daten eine normalisierte Form haben und in einer zeilen- und spaltenorientierten Datenbank gespeichert werden können. Währenddessen können unstrukturierte Daten dieses nicht. Unstrukturierte Daten besitzen eine nicht identifizierbare Datenstruktur die bewirkt, dass sie schlecht verarbeitet werden können, jedoch sind sie meistens der Ursprung der strukturierten Daten.

Diese Daten sind für Computerprogramme nicht leicht zu verarbeiten, deswegen werden diese meistens, durch verschiedenste Methoden in strukturierte Daten verwandelt. Moderne Unternehmen benutzen AI Analyse Elemente, um diese Daten zu strukturieren und zu verarbeiten. Viele Big Data Applikationen stellen nützliche Methoden bereit, um die Verarbeitung, Speicherung und Analyse der unformatierten Daten zu erleichtern.

Es ist ein Fakt, dass viel mehr unformatierte als formatierte Daten entstehen, dies hat jedoch viele verschiedene Gründe. Sogar mehr als 80 Prozent der Daten sind unstrukturierte Daten, diese bestehen hauptsächlich aus E-Mails, Bildern, Videos, et cetera (siehe Grafik unten).

Dieser Umstand ist ziemlich plausibel wenn man darüber nachdenkt, wie oft Jugendliche Bilder auf Sozialen Netzwerken posten oder E-Mails in Unternehmen genutzt werden. Obwohl unstrukturierte Daten schwer von Anwendungen bearbeitet werden können, sind sie voller nützlicher Informationen. Strukturierte Daten werden im Gegensatz zu den unformatierten Daten nicht oft erstellt, jedoch sind sie zum Verarbeiten bestens geeignet.

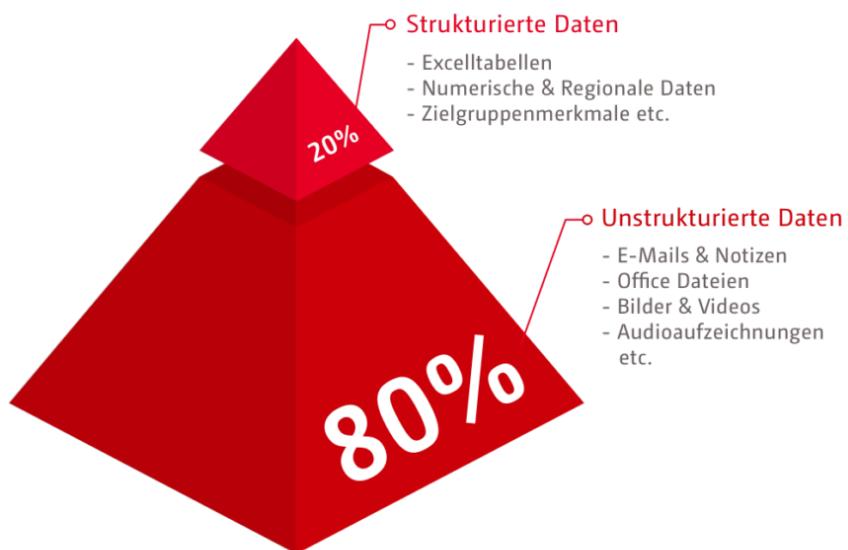


Abbildung 12.1: Datenpyramide
Quelle: (AA:Web20, vgl. neofonie.de 13.02.2021)

Abgrenzung der Daten

Im Allgemeinen werden Daten abhängig von ihrem Strukturierungsgrad nach folgenden Typen unterschieden:

- formulierte Daten
- semistrukturierte Daten
- unformulierte Daten

Im Gegensatz zu unformulierten Daten haben semistrukturierte Daten eine Grundstruktur. Ein sehr bekanntes Beispiel dafür sind E-Mails. Die E-Mail besitzt eine gewisse Grundstruktur mit Betreff, Absender und Empfänger sowie weiteren Informationen im Nachrichtenkopf, die angeführt werden können. Der Grund wieso eine E-Mail nicht zu den formulierten Daten zählt ist, dass sie zu einem Großteil aus Text besteht, der formlos ist und keine Struktur besitzt.

Formulierte Daten besitzen ein vorgegebenes Format, in das sich die Information speichern lässt. Informationen die in SQL-Datenbanken gespeichert werden zählen als formulierte Daten, da sie eine fixe Zeilen- und Spaltenposition besitzen. Der praktische Vorteil dieser Form von formulierten Daten ist es, dass sie leicht zu finden und zu bearbeiten sind.

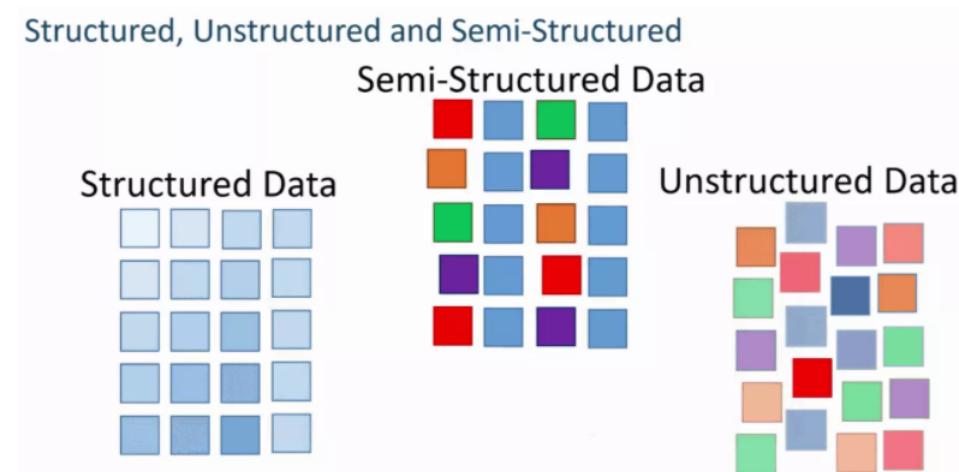


Abbildung 12.2: Arten von Daten
Quelle: (AA:Web22, vgl. astera.com 21.02.2021)

Große Unterschiede zwischen der Struktur der digitalen Daten werden hier aufgelistet:

- Organisation: Zu den organisiertesten Daten zählen die formatierten Daten. Durch ihre hohe Struktur weisen sie die höchste Organisationsebene auf, während halbstrukturierte Daten teilweise organisiert sind. Die halbstrukturierten Daten weisen deshalb eine hohe Organisationsebene auf, jedoch ist diese geringer als bei den formatierten Daten und höher als bei den unformatierten Daten. Unformatierte Daten sind überhaupt nicht organisiert.
- Flexibilität und Skalierbarkeit: Formaterte Daten sind abhängig von relationalen Datenbanken oder Schemas, daher sehr unflexibel und schwer zu skalieren. Im Vergleich dazu sind Halbstrukturierte Daten flexibler und leichter zu skalieren. Am flexibelsten und am besten zu skalieren sind unformatierte Daten, da sie nicht abhängig von Schemas oder relationalen Datenbanken sind.
- Versionierung: Es gibt viele Möglichkeiten formatierte Daten auf Basis von relationalen Datenbanken zu versionieren, die häufigsten sind Tupel, Zeilen und Tabellen. Im Gegensatz dazu benutzt man bei halbstrukturierten Daten nur Tupel oder Diagramme, da nur eine Teildatenbank diese unterstützt. Die Versionierung ist bei den unformatierten Daten als Ganzes sehr wahrscheinlich, da diese nicht von der Datenbank unterstützt wird.
- Transaktionsmanagement: Parallelität von Daten ist in formatierten Daten verfügbar und wird deshalb von Multitasking-Prozessen bevorzugt. Halbstrukturierte Daten werden normalerweise an den Transaktionen von Datenbank Management Systemen (DMBS) angepasst und deswegen ist die Parallelität nicht verfügbar. Unformatierte Daten besitzen weder Transaktionsmanagement oder Parallelität.

Daten im Unternehmen

Im Normalfall sind die meisten genutzten Daten in einem Unternehmen unformatiert, da wie zum Beispiel E-Mails täglich versendet werden, um den Datenaustausch zu ermöglichen. Für moderne Unternehmen sind das Verwalten, Speichern und Verarbeiten von unformatierten Daten eine Herausforderung der sie sich stellen müssen, da die übliche Nutzung von SQL-Datenbanken schwer realisierbar ist.

Daten können in vielerlei Hinsicht wichtig sein, deswegen versuchen Unternehmen die informativen unformatierten Daten zu analysieren, jedoch sind Verfahren notwendig, wie zum Beispiel Text- und Spracherkennung. Nach diesem Prozess können die Daten anschließend nach bestimmten Schlüsselwerten durchsucht werden. Danach können die herausgesuchten Daten aus deren Quellen entnommen werden und in relationalen Datenbanken gespeichert werden. Da sie nun als formatierte Daten in einer Datenbank

stehen, können sie dort abgerufen und durch Programme verarbeitet werden. Durch die Ergebnisse der Daten hat das Unternehmen Wissen erlangt das sie nutzen können, um Profit zu erlangen oder Optimierungen an ihren Geschäften zu machen.

Problemstellungen und Lösung

Unformatierte Daten stellen Unternehmen vor folgende Herausforderungen:

- es ist nicht möglich große Datenmengen in relationaler Form abzuspeichern und weiter zu verarbeiten
- aus unterschiedlichen Quellen verschiedenste Formen von Daten zu gewinnen, diese zu speichern und zu verarbeiten
- Anwendungen sollen performant auf die Daten zugreifen
- Speicherung und Verarbeitung der Daten soll in hoher Geschwindigkeit/Echtzeit erfolgen

Um diese Probleme zu lösen sind Techniken für große Mengen von unstrukturierten Daten von Nöten. Normale Datenverwaltungsprozesse und -programme sind für diese Menge an Daten nicht ausgelegt und deswegen nicht sehr gut zu gebrauchen.

Eine Möglichkeit ist es die Prozesse zu parallelisieren, indem man verteilte Infrastrukturen in Einsatz nimmt, diese verteilen die Aufgaben auf verschiedene Servercluster. Die Datenspeicher haben somit mehrere Vorteile wie zum Beispiel, dass diese große Datenmengen aufnehmen können und Techniken oder Funktionen zur Verfügung haben, womit sie die Daten komprimieren oder Datendoubletten¹ erkennen können.

Ein weiterer Vorteil der Nutzung von verteilten Datenspeichern und parallelisierten Prozessen ist, dass die komplette Architektur gut skaliert und für große Mengen von unformatierten Datenmengen geeignet ist. Nebenbei sollte auch eine Netzwerkinfrastruktur mit genügend Performance im Hintergrund arbeiten, damit die Daten schnell an andere Knoten weitergereicht werden können.

(AA:Web21, vgl. Dipl.-Ing. (FH) Stefan Luber 20.02.2021)

Praxisbeispiel

Am Beispiel AI Börse wird erklärt, wie die unformatierten Daten zu nützlichen formatierten Daten werden:

¹Dublette ist ein Datensatz, der redundant, d. h. mehrfach, vorhanden ist

Das Projektteam AI Börse beschäftigt sich, wie ihr Name schon sagt, mit der Börse, genauer gesagt mit Börsenprognosen die den Kurs des DAX prognostizieren. Diese Börsenprognosen werden mit dem echten Kursstand des DAX verglichen, um den Wahrheitswert der jeweiligen Börsenseiten zu bestimmen.

Es besteht jedoch die Frage, wie die unformatierten Rohtexte in formatierte Daten umgewandelt werden? Das ist ganz leicht zu erklären. Ein vollautomatisiertes Java-Programm (Crawler) sucht mittels Schlüsselbegriffe in den schon voreingestellten Webseiten Teile von Texten heraus. Diese Daten werden daraufhin in eine relationale Datenbank gespeichert. Der Inhalt der Datenbank kann dann genutzt werden, um zu ermitteln welche Website die höchste Wahrscheinlichkeit hat, einen richtigen Kursverlauf zu prognostizieren oder diese Daten auf der grafischen Benutzeroberfläche zu repräsentieren.

	id integer	grenze double precision	anlaufstelle double precision	link character varying(255)	datum date	uhrzzeit character varying(255)	grenzeerreicht integer	anlauferreicht integer
1	782	13300	13200	https://www.boerse-daily.de/boersen-nachrichten/in	2021-01-22	23:07:01	0	0
2	897	13300	13200	https://www.boerse-daily.de/boersen-nachrichten/in	2021-01-25	12:23:01	0	0
3	562	13800	13700	https://www.boerse-daily.de/boersen-nachrichten/in	2021-01-21	18:35:07	0	0
4	868	14000	14150	https://www.boerse-daily.de/boersen-nachrichten/in	2021-01-25	23:54:41	0	0
5	1453	13600	13850	https://www.boerse-daily.de/boersen-nachrichten/in	2021-01-29	20:59:13	0	0
6	767	14200	14500	https://www.boerse-online.de/nachrichten/ressort/m	2021-01-22	23:06:59	0	0
7	873	13460	13441	https://www.boerse-daily.de/boersen-nachrichten/in	2021-01-25	23:54:41	0	0
8	787	14000	14100	https://www.boerse-daily.de/boersen-nachrichten/in	2021-01-22	23:07:02	0	0
9	558	14100	14150	https://www.boerse-daily.de/boersen-nachrichten/in	2021-01-21	18:35:06	0	0
10	878	13200	12600	https://www.boerse-daily.de/boersen-nachrichten/in	2021-01-25	23:54:42	0	0
11	1463	13300	13000	https://www.boerse-daily.de/boersen-nachrichten/in	2021-01-29	20:59:14	0	0
12	563	13700	13672	https://www.boerse-daily.de/boersen-nachrichten/in	2021-01-21	18:35:07	0	0
13	564	13672	13600	https://www.boerse-daily.de/boersen-nachrichten/in	2021-01-21	18:35:07	0	0
14	795	13600	13595	https://www.boerse-daily.de/boersen-nachrichten/in	2021-01-22	23:07:03	0	0

Abbildung 12.3: Ausschnitt von Daten
Quelle: vgl. Projektgruppe AI Börse 20.02.2021

An diesem Ausschnitt kann man eine Tabelle mit formatierten Daten erkennen. Man erkennt, dass die Daten formatiert sind, da sie eine klare Struktur haben. Die Daten kommen aus einer PostgreSQL Datenbank und von den Börsenseiten die das Projektteam AI Börse angegeben hat, und wurden mittels einer Applikation in die Datenbank eingesetzt.

Jeder Datensatz den man sehen kann gehört zu einer Börsenprognose aus denen die Daten gewonnen wurden. Diese Daten sind von sehr hohem Wert da sie weiterverwendet werden können um Wissen über die Genauigkeit der Börsenseitenquellen zu erlangen oder um die Daten auf einer Webseite darzustellen.

In diesem Fall (siehe Abbildung 12.3) sind die Daten Identifikationsnummer, einzelne Grenzen der Prognosen, Anlaufstellen der Prognosen, ein Link zu der expliziten Börsenseite mit der jeweiligen Prognose, Datum/Zeit der aufgenommenen Prognose und 1

(wurde erreicht) oder 0 (wurde nicht erreicht) wenn die Grenze oder Anlaufstelle erreicht wurde.

Diese Daten werden ausgewertet und dadurch kann man erfahren, welche Seite wie oft und wie genau Recht mit ihren Prognosen hatte. Diese Richtigkeit kann mittels einer Tabelle auf der grafischen Benutzeroberfläche abgebildet werden. Durch die Darstellung dieser Daten können die Benutzer die Situation besser abschätzen und richtig investieren (wenn das ihre Intention ist).

12.2 Darstellung von Daten

12.2.1 Wieso werden Daten dargestellt?

Sowohl Programmierer, Industriearbeiter, Aktienhändler als auch Schüler benutzen Daten und stellen diese dar. Der Grund für die Darstellung der Daten ist simpel, die Daten enthalten Wissen, das in einer Tabelle oder einem Diagramm leicht für jeden ersichtlich ist. Man erkennt kurz auf einen Blick Datenzusammenhänge und kann diese interpretieren und für seinen eigenen Zweck verwenden.

Es gibt verschiedene Arten und Möglichkeiten Daten dazustellen. Statistiken und Tabellen sind Optionen um anderen Beteiligten die Zusammenhänge eines Themas zu erklären, wie zum Beispiel in Präsentationen. Solche Abbildungen sind sehr repräsentativ und leicht zu verstehen, weswegen sie auch oft benutzt werden.

Beispiele wie die verschiedenen dargestellten Daten genutzt werden können:

- Analysieren von Statistiken, um über Businessentscheidungen zu urteilen
- Um Zusammenhänge zwischen Daten zu repräsentieren
- Um Verläufe von bestimmten Produkten nachzuvollziehen
- Veranschaulichung von Tabellen, damit alles auf einen Blick ablesbar ist

Zum Veranschaulichen der Informationsgewinnung durch Diagramme werden Beispiele im Bezug auf Kursverläufe/Börsenseiten gezeigt, um zu beschreiben welche Informationen daraus gewonnen werden können:

Beispiel 1:

In diesem Beispiel wird die monatliche Trefferquote von Prognosen vom DAX von einer spezifischen Börsenseite ermittelt und daraus werden dann Schlüsse gezogen, ob man nach den Prognosen dieser Seite in bestimmte Aktien investieren sollte oder nicht.

- Die grünen Balken zeigen an, wie hoch die Trefferquote der Prognosen der spezifischen Börsenseite in dem dazu beschrifteten Monat war (in Prozent)
- Die roten Balken zeigen an, wie oft die Prognosen der spezifischen Börsenseite in dem dazu beschrifteten Monat nicht korrekt prognostiziert wurden (in Prozent)

Die Werte in den einzelnen Balken geben das Verhältnis zwischen der Summe von korrekt prognostizierten und die Summe von nicht korrekt prognostizierten Prognosen der spezifischen Webseite an (in Prozent).

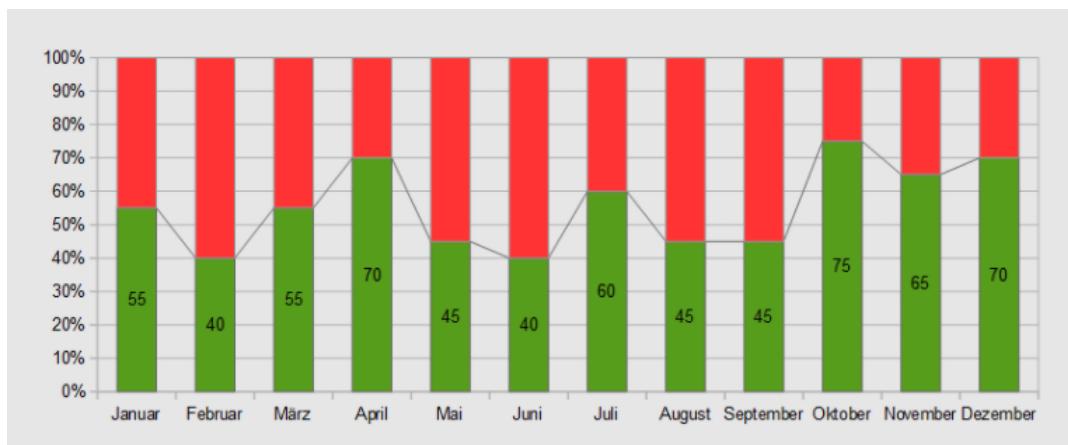


Abbildung 12.4: Trefferquote von Prognosen einer Börsenseite in Prozent
Quelle: (AA:Web27, vgl. tradistats.com 06.03.2021)

Bei genauerer Betrachtung der Grafik erkennt man Monate mit deutlichen Unterschieden zu anderen Monaten. Im Oktober beispielsweise hatte die Börsenseite 75 Prozent der Prognosen richtig geschätzt, was in diesem Fall eine Menge ist. Wenn Investoren solchen Grafiken Vertrauen schenken werden diese, bei konstant guten Resultaten, dieser Börsenseite ihr Gehör schenken und deren Prognosen folgen.

Beispiel 2:

Die unten angeführte Grafik zeigt den durchschnittlichen Gewinn/Verlust, der in den einzelnen Monaten erzielt wurde. Jedoch zeigt dieses Diagramm im Vergleich zum Beispiel 1 nicht nur die Häufigkeit, sondern auch die Höhe der Werte an. Zur Berechnung des durchschnittlichen Gewinns wurden zum Beispiel die Gewinne und Verluste, die in einem der Monate des Jahres angefallen sind, aufsummiert und dann durch die Anzahl der Jahre dividiert.

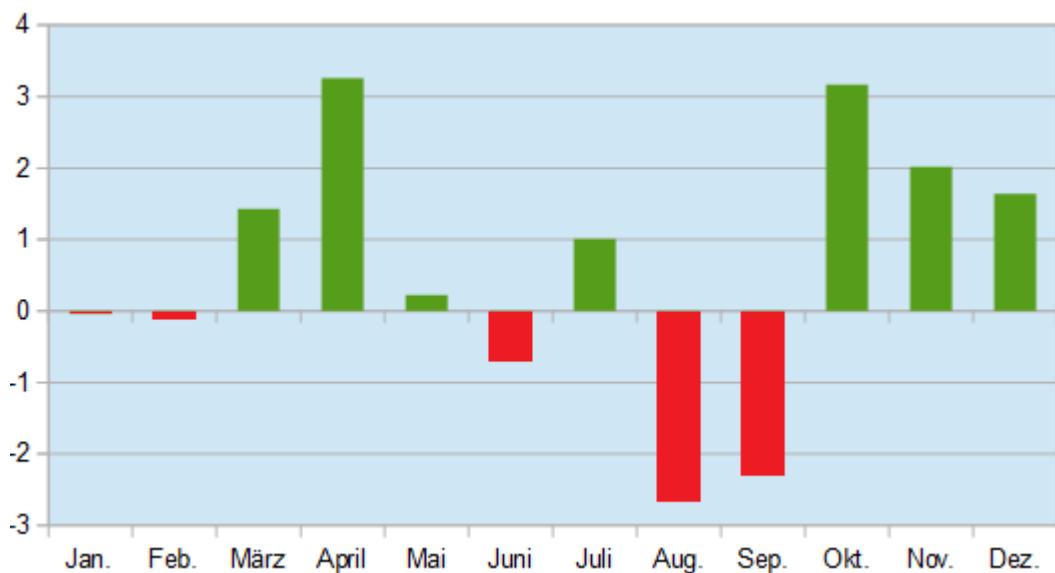


Abbildung 12.5: Durchschnittlicher Gewinn in den Börsenmonaten
Quelle: (AA:Web28, vgl. tradistats.com 06.03.2021)

Die Grafik bietet Informationen darüber, wann und wie hoch der Gewinn/Verlust war. An der Grafik kann man zum Beispiel ablesen, dass sich der DAX im August und September am schlechtesten entwickelt hat und im April und Oktober am besten.

Solche Verläufe sind besonders für Trader wichtig, da sie diese Entwicklung analysieren und anschließend darauf reagieren können. Es gibt mehrere Möglichkeiten wie es dazu hätte kommen können, Beispiele dafür sind:

- News beziehungsweise „Fake News“

Beispiel:

Kursverläufe können sowohl fallen als auch steigen, dies hängt von der Reaktion der Trader ab und ob sie durch diese Reaktion Aktien kaufen oder verkaufen. Eine angesehene Person wie Elon Musk verursacht so eine Reaktion indem er beispielsweise twittert, dass Tesla völlig bankrottgeht (in diesem Fall war diese Information ein Aprilscherz). Diese Situation führte sehr starke Kursschwankungen der Tesla-Aktie herbei. Deshalb sollten News beziehungsweise falsche News-Aussagen rund um ein Unternehmen in dieser Branche niemals unterschätzt werden.

- Erscheinen mehrerer Wettbewerber

Beispiel:

Im Jahre 2018 erschien Facebook mit der Einführung ihrer eigenen Online-Partnervermittlung. Dies machte sie zur Konkurrenz für andere Unternehmen in den USA. Beispielsweise ist die Aktie von Match Group, einem Online-Dating-Service Unternehmen, (Eigentümer von Tinder und OkCupid) um 22 Prozent gestürzt, im Gegensatz dazu stieg die Facebook-Aktie.

- sportliche Ereignisse

Beispiel:

Am Beispiel der BVB-Aktie wurde die Kursänderung analysiert. Durch die Analyse der Aktie konnte man interessante Ereignisse beobachten wie beispielsweise, dass die Aktie nach einer Niederlage des Fußballvereins gesunken und nach einem gewonnenen Fußballspiel gestiegen ist. Jedoch sind die Ergebnisse sehr liquide und können nicht bei jeder Börse gleich abgebildet werden (bei manchen sind die Auswirkungen schwächer und bei manchen stärker).

- Neuigkeiten über neue Produkte

Beispiel:

Auch neu veröffentlichte Produkte können den Kursverlauf verändern, ein Beispiel dafür ist die erste Apple Watch, die im Jahre 2014 präsentiert wurde. Das Produkt war sehr beliebt und wurde als Innovation beschrieben, dies stieß Begeisterung bei den Investoren und Fans von Apple aus. Am Tag der Veröffentlichung schoss die Apple-Aktie hinauf und fiel daraufhin sehr stark. In den nachfolgenden Tagen erholte sich die Apple-Aktie und es ging wieder bergauf.

(AA:Web30, vgl. consorenbank.de 09.03.2021)

Beispiel 3:

Um die durchschnittlich richtigen Prognosen mehrerer Börsenseiten gut abzubilden benötigt man eine Grafik wie diese. Somit können Trader die Werte auf einen Blick miteinander vergleichen und daraufhin ihre eigene Meinung bilden.

Die einzelnen Balken stellen dabei die Börsenseiten dar und diese haben eine bestimmte durchschnittliche Trefferquote von richtigen Prognosen pro Monat, die in Prozent gemessen wird. Das heißt je höher der Balken ist, desto öfter hatten diese richtige Prognosen.

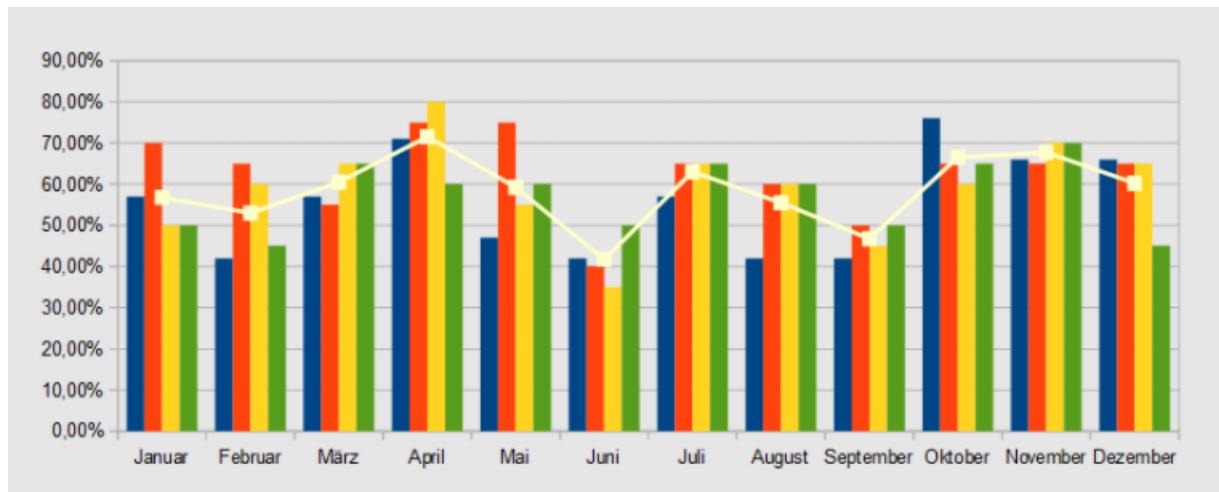


Abbildung 12.6: Durchschnittliche Trefferquote von richtigen Prognosen
Quelle: (AA:Web29, vgl. tradistats.com 06.03.2021)

Wie man sieht variiert der Wert von Börsenseite zu Börsenseite, da sie verschiedene Methoden haben die Daten auszuwerten, jedoch lässt sich ein Trend beobachten. Die Börsenseite mit dem orangenen Balken hält sich meistens über dem Durchschnitt auf was darauf schließen lässt, dass die Kursverläufe relativ oft richtig vorhergesagt werden.

Das waren ein paar Beispiele zur Darstellung im Bezug zu Kursverläufen/Börsenseiten, jedoch gibt es auch viele weitere Arten Daten darzustellen, beispielsweise in Präsentationen. Jedes dieser Hilfsmittel hat seine Vor- und Nachteile, die bei der Auswahl berücksichtigt werden sollten.

Die Auswahl der richtigen Darstellungsformen ist Situations- und Gemütsabhängig und sollte gut überlegt sein, denn jede der Grafiken hat seine Vor- und Nachteile. Nicht aufgelistete („dennoch sehr praktische“) Hilfsmittel sind:

- Flächendiagramm
- Netzdiagramm
- Häufigkeitstabellen
- Liniendiagramme

- Kreisdiagramme (Tortendiagramme)

12.3 Grafische Benutzeroberfläche

12.3.1 Definition einer GUI

Noch ein sehr wichtiger Punkt für die Darstellung von Daten ist die grafische Benutzeroberfläche, auch GUI genannt. In der heutigen Zeit ist es gang und gäbe grafische Benutzeroberflächen zu benutzen, während vor einer nicht allzu langen Zeit statt GUIs Kommandozeilen als Standard galten.

Um euch das Thema etwas näher zu bringen, wird die grafische Benutzeroberfläche hier etwas genauer erklärt:

Eine GUI, Abkürzung für Graphical User Interface, ist ein Computerprogramm, mit dem eine Person mithilfe von Symbolen, visuellen Metaphern und Zeigegeräten mit einem Computer kommunizieren kann.

Beste Beispiele für die Umsetzung sind die Oberflächen der Betriebssysteme von Apple und Microsoft. Die GUI ist die Standard-Schnittstelle heutiger Computer. Sie ersetzt die oft nur mit schwer zu merkenden Befehlen nutzbaren Text-Schnittstellen durch ein relativ intuitives System.

(AA:Web31, Jesko 16.03.2021)

Die grafische Benutzeroberfläche ist die Schnittstelle zwischen Software und Anwender, steuert die Applikation und ruft deren Funktionen auf. Sie ist ansehnlich und bietet dem Benutzer eine leicht verständliche Bedienungssoberfläche, um auch nicht so technisch begabten Nutzern die Steuerung der Software zu erleichtern.

12.3.2 Funktion einer GUI

Da geklärt wurde was eine grafische Benutzeroberfläche ist sollte man auch wissen, wie sie funktioniert und welche Möglichkeiten es gibt, sie zu bedienen. Jede GUI unterscheidet sich von anderen und hat andere Designpatterns, jedoch haben sie den gleichen allgemeinen Zweck. Die Funktion und die Nutzungsmöglichkeiten der GUIs wird hier kurz verdeutlicht:

Das GUI ist eine Benutzeroberfläche, die es Usern erlaubt, mit dem Computer zu kommunizieren. Üblich ist vor allem die Interaktion per Maus und

Tastatur (wobei inzwischen auch die Steuerung über Gesten immer mehr an Bedeutung gewinnt): Wenn wir eine Computermaus bewegen, dann bewegt sich der Zeiger auf unserem Bildschirm. Das Signal des Gerätes wird an den Computer übermittelt, der es dann in eine ähnlich verlaufende Bewegung auf dem Bildschirm übersetzt. Wenn ein Nutzer dann zum Beispiel auf ein bestimmtes Programmsymbol in der Menüauswahl klickt, wird entsprechende Anweisung ausgeführt und das Programm wird geöffnet.

Das GUI ist also eine Art Übersetzer in der Kommunikation zwischen Mensch und Maschine. Ohne GUI müsste man Programme und Anwendungen über Befehle in der Kommandozeile aufrufen
(AA:Web34, ionos.de 20.03.2021)

Seitdem es grafische Benutzeroberflächen gibt, werden sie für die verschiedensten Zwecke benutzt. Beispielsweise kann man GUIs dazu verwenden Daten zu erstellen, konfigurieren, löschen, auszugeben und verschiedene Funktionen aufzurufen, ohne auf eine umständliche Kommandozeile zuzugreifen.

In der unten abgebildeten Grafik sind die einzelnen Schichten einer GUI abgebildet. Die bestehenden Schichten sind Benutzer, grafische Benutzeroberfläche (GUI), der Anzeigeserver mit dem Fenstermanager, der Kernel und die Hardware. Diese Schichten arbeiten miteinander und werden gebraucht, um die Daten auf die GUI zu bringen.

Kurz zusammengefasst ist die GUI-Schicht sowohl für die Darstellung der Daten, als auch für die Dialogführung zuständig. Nach den Eingaben und Ausgaben der Benutzeroberfläche kommt es zur Aktion der Applikation und Methoden werden aufgerufen. Hinter der GUI könnten Datenbanken stehen, welche die eingegebenen Daten aus der GUI speichern oder Daten auf der grafischen Benutzeroberfläche darstellen.

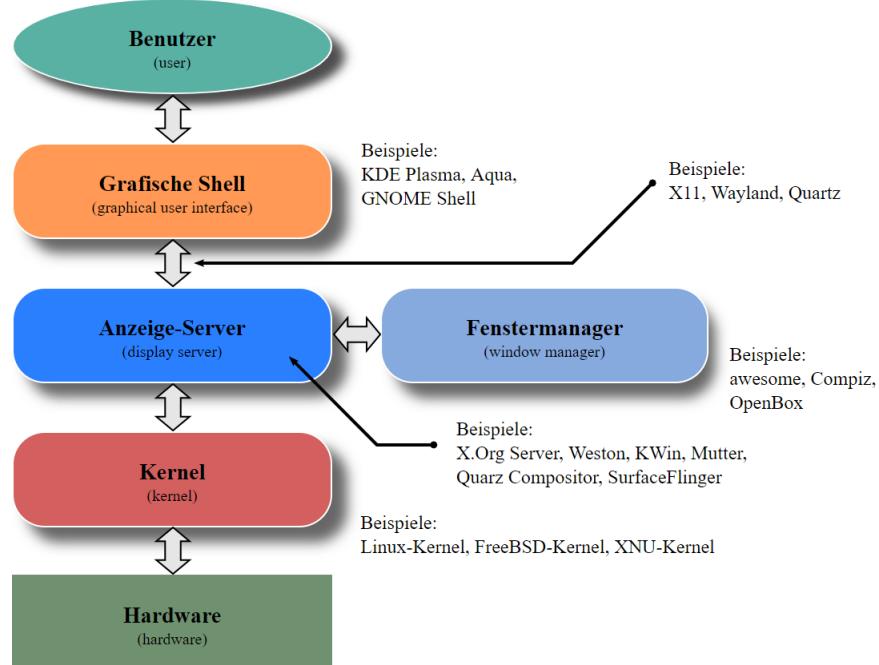


Abbildung 12.7: Schichten einer GUI
Quelle: (AA:Web37, vgl. thuatngumarketing.com 20.03.2021)

Anforderungen

Das Hauptziel einer GUI ist es, dem Nutzer die Bedienung zu erleichtern und deswegen ist es wichtig zu wissen, was man in eine GUI zu implementieren hat. Die Bedürfnisse des Kunden beziehungsweise Nutzers stehen an oberster Stelle, deswegen sollte man auch ein gutes Verständnis dafür besitzen. Man sollte ein Design wählen das die Ziele möglichst gut erfüllt und auf den Kunden zugeschnitten ist.

Bei der Entwicklung einer GUI gibt es so manche Anforderungen und Merkmale auf die man achten muss, sehr wichtige werden hier aufgezählt.

Worauf man bei der Entwicklung eines GUIs auf jeden Fall achten sollte, sind folgende Merkmale:

- *Einfache Schnittstellen: Es empfiehlt sich, auf unnötige Design-Elemente zu verzichten und einfache und klar verständliche Bezeichnungen zu wählen.*
- *Zweckorientiertes Layout: Jede Seite sollte gut strukturiert sein; jedes Element sollte eine klare Funktion erfüllen.*

- *Konsistenz: Bei der Verwendung mehrerer Elemente und Grafiken ist es wichtig, dass alle einzelnen Bestandteile aufeinander abgestimmt sind.*
- *Design und Typografie: Design-Einheiten, Farben und Texte sollten je nach Zweck eines Bestandteils das Element in den Vordergrund oder Hintergrund rücken. Wichtig ist auch, auch jeweils zur Funktion und passende und leicht erkennbare Fonts und Schriftgrößen zu verwenden.*
- *Benutzer Updates: Ein GUI sollte auch über Fehler und Status-Änderungen informieren.*

(AA:Web35, ionos.de 20.03.2021)

Wie gerade schon aufgezählt wurde hat man viele Aufgaben zu bewältigen, jedoch sollte man nicht vergessen, dass es weitere Merkmale gibt die man zu erfüllen hat und diese unterscheiden sich von den Nutzern und dem Zweck. Die Aufzählung ist immerhin ein guter Leitfaden, nach dem man sich als Arbeiter richten kann und man sollte auf der sicheren Seite stehen, wenn man diese Punkte erfüllt hat.

Bestandteile einer grafischen Benutzeroberfläche

Nicht jede grafische Benutzeroberfläche bietet die gleichen GUI-Komponenten an, dennoch gibt es oft Trends beziehungsweise ähnliche Angehensweisen, wie eine GUI aufgebaut werden kann. Anstatt alle GUI-Komponenten zu nennen werden hier die häufigsten aufgelistet:

- **Textfelder:**
Statischer Text der nicht vom Benutzer verwendet werden kann, dient als Angabe von Handlungsanweisungen und Beschriftung von Eingabefeldern.
- **Eingabefelder:**
Bereich zur Eingabe von Daten. Dieser Bereich kann gefiltert werden, um falsche Eingaben abzuweisen.
- **Schaltflächen:**
Ermöglicht den Nutzern durch Anklicken die hinterlegte Funktion aufzurufen. Meistens wird die Schaltfläche mit einem Text beschriftet.
- **Optionsfeld (Checkbox):**
Betitelt einen Bereich der ein- oder ausgeschaltet werden kann.

- Dropdown-Liste:
Dient dazu eine Auswahl zu treffen, die nach einem Klick aufgelistet wird.
- Lokales Menü: Beinhaltet mehrere Funktionen, die sich auf eine bestimmte Komponente auf der grafischen Benutzeroberfläche beziehen.

Um die richtigen GUI-Komponenten zu benutzen oder auszuwählen muss man sich Gedanken machen, welche Funktionen und welche Eingaben/Ausgaben die GUI beinhalten soll. Ohne einen konkreten Plan kann man keine passende GUIs erstellen. GUI werden meistens nach den Funktionalitäten programmiert, somit ist man flexibel und kann seiner Kreativität freien Lauf lassen.

Ein sehr wichtiges Merkmal ist das Aussehen der einzelnen Elemente. Der Benutzer sollte beim ersten Anblick der Komponente direkt erkennen können was die Funktion ist. Ein gutes Beispiel zu diesem Merkmal ist der Papierkorb, denn dieses Element wird bei mehreren Betriebssystemen benutzt, beispielsweise in Windows und Mac. Den Nutzern ist klar wozu der Papierkorb dient: Löschen von Dateien.

Es ist üblich eine GUI ereignisbasiert zu entwickeln da man nicht voraussagen kann, welche Aufgabe der Anwender als nächstes ausführt. Deswegen programmiert man eine GUI nicht linear, sondern schreibt sie so, dass auf das Signal des Users reagiert wird, wenn er ein Singal über gibt.

(AA:Web36, vgl. ionos.de 20.03.2021)

12.3.3 Design und Qualität einer GUI

Jede GUI unterscheidet sich von der anderen und hat andere Ziele und Schwerpunkte. Ein Programm das beispielsweise im Büroalltag gebraucht wird hat andere Ansprüche, als die grafische Benutzeroberfläche einer Dokumentationssoftware in einer Kfz-Werkstatt.

Das Erscheinungsbild kann Abzüge an der Funktionalität der Software mit sich tragen, deshalb sollte man gut einschätzen welche grafische Komponenten man genau implementiert und welche nicht. Im besten Fall wiedersprechen sich Design und Funktionalität nicht, sondern harmonieren miteinander. Eine Anwendung mit einer grafischen Benutzeroberfläche lässt den Anwender vergessen, dass er ein Werkzeug bedient.

Je nach Anwendung wird mehr Augenmerk auf Performance oder grafisches Erscheinungsbild gelegt, beispielsweise wird, wenn eine Anwendung geschrieben wird um Daten darzustellen, sehr viel Wert auf das Design gelegt, damit die Anwender leicht wichtige Daten herauslesen kann. Auf der anderen Hand wird bei einer Software die

viele Algorithmen verwendet und wenige grafische Komponenten benötigt sehr großer Wert auf Performance gelegt.

Abgesehen von der Art des Designs gibt es eine Menge Merkmale, die eine gute Anwendung ausmachen. Das wörtliche Zitat erklärt wie es zu einem guten Programm kommt, im Detail:

Nach allgemeiner Auffassung ist eine Software von hoher Qualität, wenn

- 1. die benötigte Funktionalität bereitgestellt wird, d. h. die geforderten Funktionen werden beispielsweise mittels Anwendungsfälle im Pflichtenheft dokumentiert.*
- 2. bestimmte technische und messbare Anforderungen erfüllt sind. Dazu gehören u. a. die Verarbeitungskapazität, der Datendurchsatz oder die Geschwindigkeit der Verarbeitung.*

Diese Punkte sind Merkmale zur Feststellung der objektiven Qualität einer Software. Der wichtigste Aspekt ist jedoch: „Der Anwender muss die Software für gut halten“. Demnach ist also die subjektive Qualität entscheidend. (AA:Web32, Dr. Veikko Krypczyk 16.03.2021)

Wie schon oft zählt die Meinung des Kunden am meisten, denn der entscheidet, ob das Programm zu gebrauchen ist oder überarbeitet gehört. Es gibt mehrere Faktoren die entscheiden, ob der Nutzer auf lang- und mittelfristiger Zeit zufrieden gestellt ist oder ob das Programm ein kompletter Reinfall war. Zu diesen Faktoren gehören:

- Die Anwendung funktioniert und man kann seine Arbeit damit erledigen.
- Man erhält Support, wenn etwas unklar sein sollte.
- Kein Datenverlust! Daten über Personen müssen gut aufbewahrt werden und es ist sorgsam mit ihnen umzugehen. Ein nicht vorhergesehener Datenverlust könnte Probleme mit dem Kunden bedeuten und die Arbeit verliert jegliches Vertrauen.
- Ungewünschte Nebeneffekte sollen nicht auftreten (Performance-Probleme, Bugs, Laden von Daten/Bildern...).
- Um seriöse Arbeit vollbringen zu können, muss man dem Programm bei Normalgebrauch vertrauen nicht abzustürzen oder größere Probleme zu bereiten.

Trends zum GUI-Design

So wie vieles andere entwickeln sich grafische Benutzeroberflächen immer weiter. Jedoch erscheinen immer wieder bestimmte Ähnlichkeiten zwischen den GUIs. Stichwörter wie minimalistisch reduziert und kontextbezogen, beschreiben die aktuellen Trends der Benutzeroberflächen sehr gut. Sehr populäre Trends in Desktop-Applikationen sind:

Funktionaler Minimalismus:

Trend beschränkt die Benutzeroberfläche, wie der Name schon sagt, auf ein Minimum und stellt nur wesentliche Punkte dar. Die Navigation auf der Oberfläche wird schlicht und einfach gehalten. Mehrfach verschachtelte Menüpunkte werden als misslungen betrachtet und der gewonnene Platz wird dazu verwendet Inhalte einzufügen, wie zum Beispiel Diagramme oder Charts.

Stilistischer Minimalismus:

legt Wert auf das schlichte Aussehen der GUI. Das Ziel ist es die grafische Benutzeroberfläche mit wenigen Farbtönen zu verzieren und wichtige Aspekte mit Akzenttönen hervorzuheben. Das Augenmerk liegt auf Inhalt und nicht auf einer schönen Benutzerumgebung.

Integrierte Bedienelemente:

umfasst das vereinfachen der Designelemente, beispielsweise werden Menüs und Symbolleisten kombiniert, dadurch entstehen sogenannte Multifunktionsleisten.

Konfigurierbare UI:

man soll die Oberfläche flexibel gestalten, damit der Nutzer so arbeiten kann, wie er will. Es werden möglichst viele Anzeigeelemente frei platzierbar und können an andere Elemente angedockt werden. Durch diese Funktionen der GUI, kann der Benutzer alles auf seine Wünsche anpassen.

Sensorische Bedienelemente:

es wird versucht möglichst oft eine Interaktion mittels Touch-Eingabe zu erschaffen. Früher war das eine sehr große Innovation, da dieses nur über Touchpad (Laptop) oder Smartphones möglich war, jedoch wird dieser Trend regelrecht bei allem verwendet, wie beispielsweise an externen Monitoren, Notebooks und vielem mehr.

Es werden auch haptische Interaktionen realisiert, das heißt Bedienung mittels Gesten. Ein wesentliches Merkmal dieses Trends ist, dass der Inhalt selbst als Navigationselement dient.

Multi-Device-UI:

dient dazu die UI an die verschiedensten Displaygrößen anzupassen. Die selbe Software soll gleich an verschiedenen Geräten laufen, egal ob auf Smartphones mit einem kleinen Display oder an einem externen Monitor.

Designsprachen

Skeuomorphismus: Diese Methode wird benutzt um Dinge aus der realen Welt nachzuahmen. Beispielsweise werden Applikationen geschrieben, die sich auf Bücher konzentrieren. Somit werden zwar nachgeahmte, jedoch ziemlich realistische Bücher angezeigt, die dann angeklickt werden können und eine Funktion auslösen. Diese Art und Weise der Bedienung ist klar auf dem Rückzug.

Flat Design: Im Gegensatz zu Skeuomorphismus verzichtet Flat Design komplett auf die Nachahmung der Realität. Das Hervorheben durch Farben und Effekte wird sparsam benutzt, da diese Methode auf schlichte Eleganz zielt.

Modern UI: Moderne grafische Benutzeroberflächen haben besondere Merkmale, diese sind zum Beispiel dunkle Hintergründe, einfarbige Symbole und helle Schriften. Diese Methode entstand mit der Zeit, da diese an Popularität gewonnen hat. Viele GUIs werden in diesem Stil erstellt oder bieten zumindest eine Funktion um zwischen den Designs zu wechseln.

Responsive Design: Responsive Design ist sehr flexibel, wenn es um die Ausgabe von Inhalt auf verschiedenen Geräten geht. Das Design wird dem Ausgabemedium angepasst und ist somit leicht zu lesen beziehungsweise bedienen. Applikationen haben oftmals das Problem mit der Ausgabe von Inhalten und dieses Design ist eine gute Lösung dafür, jedoch muss man beachten, dass der Inhalt bei jedem Gerät anders aussehen kann.

Material Design: Das Ziel dieser Designmöglichkeit ist es, eine gute Gestaltung mit modernen technischen Möglichkeiten zu mischen und somit eine für den Benutzer freundliche und schöne Benutzeroberfläche zu bieten.

Dieses Design wird üblicherweise in kartenähnlichen Flächen dargestellt und beinhaltet Ähnlichkeiten mit der Designsprache Flat Design. Noch dazu ist Material Design bekannt für seinen Minimalismus, dennoch verwendet man Animationen und Schatten, um dem Nutzer zu zeigen welche Elemente bedienbar sind und welche nicht.

(AA:Web33, vgl. Dr. Veikko Krypczyk 16.03.2021)

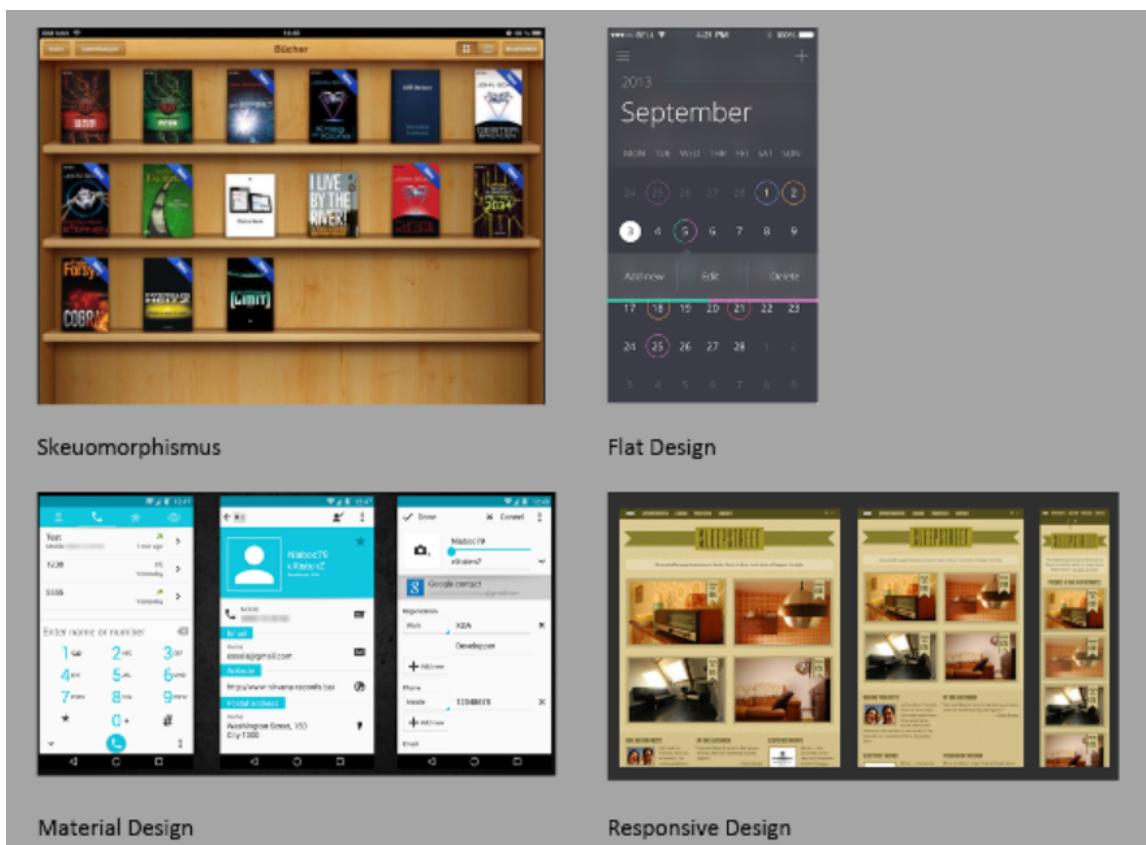


Abbildung 12.8: Designsprachen
Quelle: (AA:Web38, vgl. entwickler.de 20.03.2021)

Vor- und Nachteile

Die grafische Benutzeroberfläche zählt nicht ohne Grund als ein revolutionärer digitaler Meilenstein der Informatik, denn die GUI bietet mehrere markante Vorteile mit sich. Abgesehen von den Vorteilen, gibt es jedoch auch Nachteile die man im Hinterkopf behalten sollte, wenn man eine GUI erstellt.

Alle Pros und Contras bezüglich der grafischen Benutzeroberfläche werden hier kurz aufgelistet:

Pro:

- Leicht zu bedienen und besonders freundlich für den Anwender
- Schön gestaltet und optisch ansprechend
- Simple Anwendungen sind über die GUI leicht zu bedienen, auch für Nutzer ohne viel technischem Know-How
- Umständliches Suchen von Daten und Dokumenten wird durch die optische Darstellung deutlich erleichtert
- Durch die grafische Oberfläche kann der Nutzer die Eingaben ganz intuitiv eingeben, worauf das System anschließend reagiert
- Die Navigation zwischen den Anwendungen wird durch die grafische Oberfläche sehr einfach und schnell

Contra:

- Weniger flexibel, da nur vorprogrammierte Anweisungen genutzt werden können
- Funktionalitäten der GUI können nicht verändert werden
- Benötigt viel Speicherkapazität
- Das Design der GUI ist im Vergleich zur Kommandozeile schwer zu entwickeln
- Funktionen könnten über die grafische Oberfläche mehr Zeit benötigen

(AA:Web39, vgl. ionos.de 22.03.2021)

Diese Vor- und Nachteile sollte man im Blick haben, wenn man eine GUI entwickeln will und man sollte abschätzen können, welche Merkmale im Projekt mehr gebraucht werden. In den meisten Fällen ist es klüger eine GUI für eine Applikation zu erstellen, wenn Kunden von dieser Gebrauch machen.

12.4 Vergleich mit dem Projekt AI Börse

Die Projektgruppe AI Börse hat sich als Ziel gesetzt eine funktionsfähige, leicht bedienbare und optisch ansprechende grafische Oberfläche zu entwickeln. Auf der grafischen Oberfläche werden bereits ausgewertete Daten der Börsenprognosen angezeigt. Die Oberfläche sollte bis zum Ende des Projektes funktionsfähig sein und die richtigen Daten ausgeben.

12.4.1 Planung

Das Entwickeln einer grafischen Benutzeroberfläche ist keine einfache Arbeit und benötigt eine Menge Zeit. Um eine funktionsfähige und gut gestaltete GUI auf die Beine zu stellen bedarf es einer gründlichen Planung. Ohne einer passenden Planung ist es sehr schwer, den Kunden ein gutes Endprodukt zu präsentieren.

Beschreibung

Durch das Verbinden mit unserem Webserver über einen geeigneten Browser, wird die grafische Benutzeroberfläche angezeigt. Dem Benutzer wird angezeigt, wann der letzte Crawlingvorgang ausgeführt wurde, sprich, wie aktuell die gezeigten Daten sind. Der Name der Börsenseite, und die Bewertung der Prognosen (in Prozent), werden in der Bewertungs-Tabelle dargestellt. Die Bewertungen werden für die letzten 7 Tage, 30 Tage, und für einen optionalen Zeitraum, welcher mit einem Drop-Down-Menü einstellbar ist, angezeigt.

[siehe Pflichtenheft des Projektes AI Börse]

Dieser Absatz ist ein kurzer Auszug aus dem Pflichtenheft der Projektgruppe AI Börse und dazu zählt dieser die Kernaufgaben der grafischen Benutzeroberfläche auf. Diese Funktionen müssen in die GUI implementiert werden, um als erfolgreich und gut gelungen angesehen zu werden und die Ziele der GUI sollten keine markanten Abweichungen aufweisen.

Die Planung beinhaltet auch die Überlegung, wie die Daten auf der Webseite angezeigt werden. In diesem Fall wurde eine Skizze angefertigt, nach der sich die Projektgruppe orientiert hat, jedoch ist diese Skizze nicht gleich das Endprodukt, da man möglicherweise Funktionen adaptieren muss oder Hindernisse auftauchen, die man umgehen sollte.

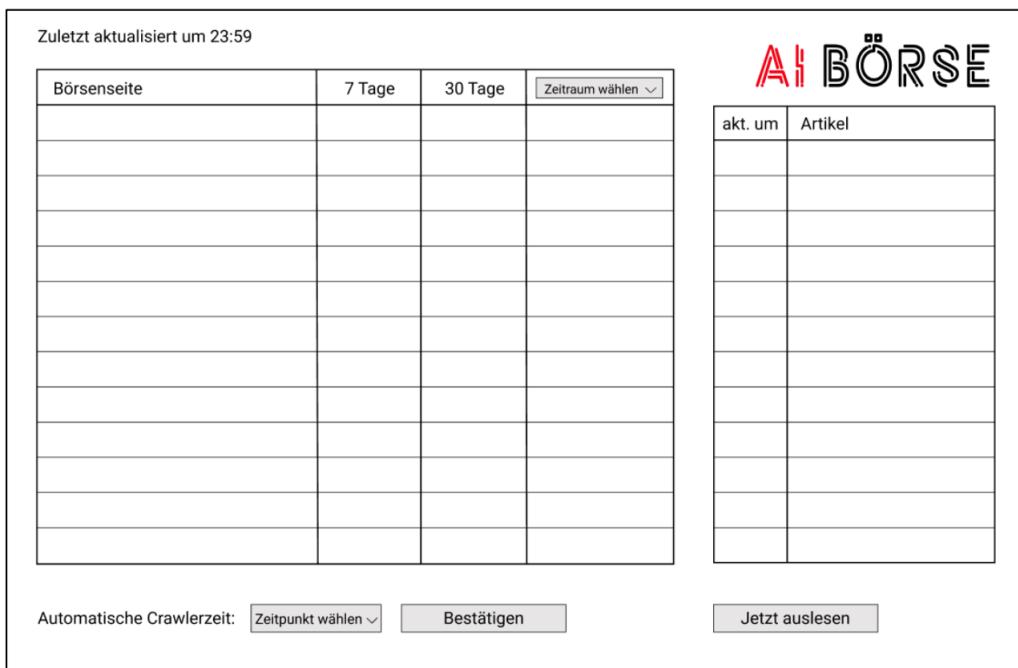


Abbildung 12.9: Skizze der GUI
Quelle: Projektgruppe AI Börse

Komponenten der Skizze

Geplant war eine GUI mit zwei Tabellen, zwei Drop-Down-Menüs, zwei Buttons, dem Logo der Projektgruppe und Textfelder zum Beschriften der Tabellen, Steuerelementen der GUI und zum Darstellen der Uhrzeit (siehe Abbildung 12.9).

Die geplanten Funktionen der GUI werden nun hier im Detail beschrieben:

Bewertungs-Tabelle:

Es werden alle Quellseiten der ausgewählten Börsennewsseiten angezeigt und neben ihnen wird in Prozent präsentiert, wie oft die einzelnen Börsennewsseiten in den angegebenen Zeiträumen richtige Prognosen getroffen haben.

Zeitraum Drop-Down-Menü:

Dieses Steuerelement liegt im Tabellenkopf der Bewertungs-Tabelle und wird dazu benutzt den Zeitraum auszuwählen, in der die Daten angezeigt werden sollen.

Mögliche Werte sind: 15 Tage, 60 Tage, 80 Tage, 180 Tage, 365 Tage

Log-Tabelle:

Diese Tabelle zeigt die letzten hinzugefügten Prognosen mit der Zeit des Hinzufügens (Timestamp) an.

Zeitpunkt Drop-Down-Menü:

Dieses Drop-Down-Menü steuert die Zeiten in denen der Crawler nach weiteren Artikeln sucht.

Mögliche Werte sind: 8 Uhr, 10 Uhr, 12 Uhr, 14 Uhr, 16 Uhr, 18 Uhr, 20 Uhr

„Bestätigen“-Button:

Dieser Button bestätigt die neue Auswahl im Zeitpunkt Drop-Down-Menü.

„Aktualisieren“-Button:

Dieser Button löst manuell einen Suchvorgang am Crawler aus.

Diese Informationen wurden als Leitfaden für das Projekt genutzt. Die Ergebnisse vom Endprodukt könnten von der Skizze abweichen, da sich das Projektteam beim Start des Projektes nur grobe Informationen zu den Börsenbriefen aneignen konnte und sie die später auftretenden Probleme nicht genau vorhersagen konnten.

12.4.2 CodeIgniter

12.4.3 Was ist CodeIgniter?

CodeIgniter ist ein PHP-MVC-Framework² das hilft, möglichst schnell eine Webanwendung zu entwickeln. Es bietet einsatzbereite Bibliotheken zum Herstellen einer Verbindung zur Datenbank und mit CodeIgniter ist es möglich verschiedene Vorgänge auszuführen, beispielsweise Senden von E-Mails, Hochladen von Dateien oder Verwalten von Sitzungen.

CodeIgniter bietet viele vorteilhafte Eigenschaften die einem beim Erstellen der Webanwendung helfen, ein größer Teil von ihnen wird hier erklärt, damit man einen kleinen Einblick in das Framework bekommt:

Kleine Datenmenge

Einer der vielen Vorteile ist die Größe der Quellcodes von CodeIgniter-Framework, denn diese liegt bei knappen 2 MB. Diese Eigenschaft trägt dazu bei, dass der Nutzer CodeIgniter und seine Funktionsweise leichter beherrschen kann. Nebenbei werden die Bereitstellung und Aktualisierung durch die Größe des Quellcodes erleichtert.

Schnelle Ladezeiten

Im Vergleich zu anderen modernen Frameworks braucht CodeIgniter durchschnittlich 50 ms zum Laden der Daten. Andere Frameworks nutzen die Zeit für die Optimierung,

²MVC steht für Model View Controller

jedoch wird diese frei, wenn man mit dem CodeIgniter-Framework arbeitet.

Leicht verbunden

Das Framework wurde so konzipiert, dass mehrere Funktionen parallel und unabhängig voneinander ablaufen können. Dies erleichtert die Durchführung und das Hinzufügen von Erweiterungen.

MVC-Architektur

Das CodeIgniter-Framework verwendet das Architekturentwurfsmuster Model-View-Controller. Dies ist ein momentaner Standard bei der Arbeit mit Webanwendungen. Die Daten, Geschäftslogik und Präsentation werden vom MVC getrennt.

Hervorragende Dokumentation

Ein sehr großer Vorteil wenn man mit CodeIgniter arbeiten will ist, dass die Dokumentation sehr umfangreich ist. Das heißt sehr viele Methoden sind beschrieben und es gibt viele Beispiele, wie man etwas machen kann. Noch dazu gibt es gute Tutorials, Bücher und beantwortete Forumfragen.

Anwendungsspezifische integrierte Komponenten

CodeIgniter besitzt eine Vielzahl von Komponenten die dazu da sind, E-Mails zu senden, Datenbanken zu verwalten, Sitzungen zu verwalten und vieles mehr.

Erweiterbar

Ein guter Helfer sind die bereits enthaltenen Bibliotheken, die einem jederzeit zur Verfügung stehen und verwendet werden können. Wenn eine Methode oder ein Code-Teil nicht vorhanden ist, könnten sie diese mittels einer Bibliothek einfach und praktisch einfügen. Nebenbei kann man auch eine REST-API im CodeIgniter-Framework erstellen.

Kurze Lernkurve

Das Framework ist leicht zu erlernen und besonders einfach für Programmierer die schon Erfahrung mit PHP gemacht haben. In kürzester Zeit können Anfänger relativ professionelle Anwendungen in CodeIgniter schreiben.

Funktionsweise von CodeIgniter

Die Funktionsweise des Frameworks ist relativ leicht zu erklären. Um es möglichst anschaulich zu erläutern, wird die Funktionsweise mittels eines Beispiels und einer dazugehörigen Darstellung präsentiert:

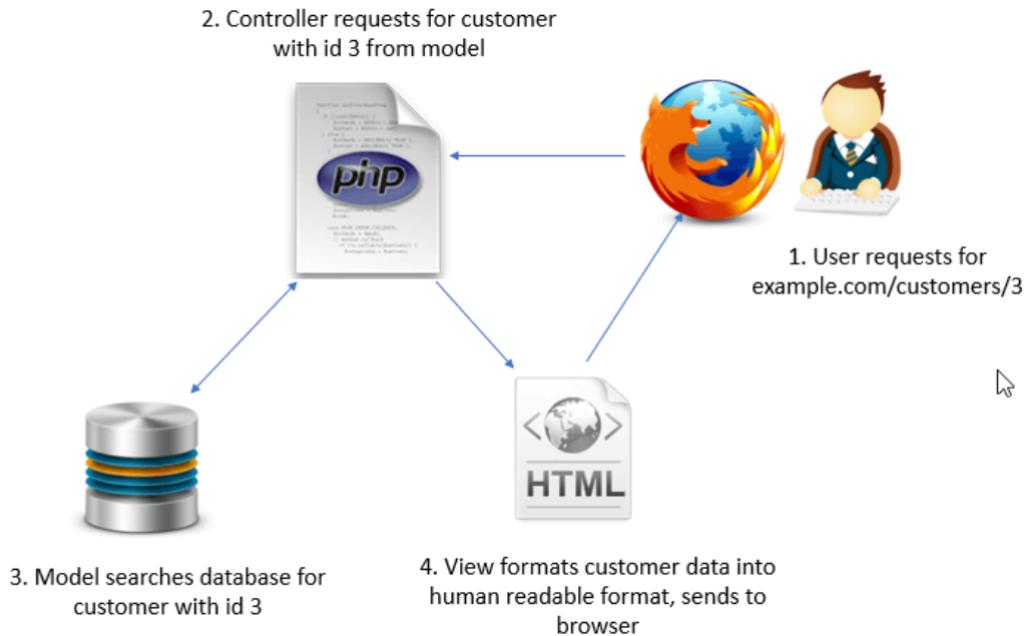


Abbildung 12.10: Funktionsweise von CodeIgniter
Quelle: (AA:Web41, vgl. guru99.com 23.03.2021)

Man will einen Kunden mit der Kunden-ID = 3 abrufen, dabei erhält der Controller die Anfrage und fordert die CodeIgniter-Modelle auf, den Datensatz mit der ID = 3 abzurufen. Die Modelle des Frameworks retournieren den Datensatz an den Controller, dieser leitet das Resultat dann an die Ansicht weiter. Die Controller formatieren dann das Resultat in ein vom Nutzer lesbaren Format um und anschließend wird das Ergebnis im Browser an den Nutzer zurückgegeben.

(AA:Web40, vgl. guru99.com 23.03.2021)

12.4.4 Implementation der GUI

Die Implementation der grafischen Benutzeroberfläche war ein langer Prozess, der sich über Monate gestreckt hat. Der Grundstein dafür war die Planung, die in den vorigen Kapiteln erklärt wurde.

Nach der Planung hat sich das Projektteam AI Börse Gedanken gemacht, wie die Skizze realisiert werden könnte und beschloss nach ausgiebigem testen der Möglichkeiten in den ersten Wochen der Implementationsphase, das CodeIgniter-Framework dafür zu benutzen. Die Gründe dafür wurden im letzten Kapitel ausführlich erklärt.

Durch das neu ausgewählte Framework hat es mehrere Wochen gedauert, um brauchbare Ergebnisse zu erbringen. Weitere Gründe waren die mangelnden Fertigkeiten in den Programmiersprachen und Entwicklungstechniken. Diese Hindernisse wurden nach dem Einlesen in die Handbücher und Aneignen der Techniken mittels Tutorials und praktischer Übung gelöst.

Der Anfang der Implementation begann mit der Grundstruktur der GUI. Das heißt es wurden alle Steuerelemente wie Buttons, Tabellen, Textfelder usw. erstellt und positioniert, dabei wurde drauf geachtet, dass das Aussehen der Skizze möglichst gut nachgeahmt wird. Die Tabellen wurden mit Testdaten gefüllt, da die realen Daten noch keine Schnittstelle zur grafischen Oberfläche hatten.

Nach der Grundstruktur wurden Programme entwickelt, die als Schnittstelle zwischen der GUI und der PostgreSQL fungierten, somit war es möglich die echten Daten aus der Datenbank zu nutzen und darzustellen.

Nachdem die Tabellen gefüllt wurden wurde begonnen, die Funktionen auszuprogrammieren, jedoch kamen währenddessen neue Anweisungen und bessere Vorschläge, die GUI zu realisieren, somit wurden die Elemente in der grafischen Oberfläche neu gestaltet, entfernt, verändert und verbessert.

Die GUI wurde am laufenden Band geändert, da neue Funktionalitäten gebraucht wurden, beispielsweise erkannte man, dass Börsenprognosen auch eine 2. Anlaufstelle beschreiben konnten und diese mussten natürlich auch auf der GUI dargestellt werden. Solche Korrekturen kamen mehrmals vor, wodurch sich die Arbeit entsprechend in die Länge zog.

Schlussendlich wurde eine Struktur festgehalten, die man dann auch programmierte und mit den Daten aus der Datenbank füllte. Parallel zu diesen Arbeiten wurde ein Linux-Server aufgesetzt der über einen Apache-Webserver die CodeIgniter-Applikation online darstellt.

Die wahrscheinlich schwerste Arbeit für das Projektteam, war das hinzufügen der Daten für die verschiedenen Zeiträume. Nachdem dies geschafft war, wurde die Oberfläche finalisiert.

Nachdem alles erreicht wurde beobachtete man die Ergebnisse und korrigierte diese, wenn Fehler erschienen sind. Somit wurde die GUI rechtzeitig fertig gestellt und es gab keine Verzögerungen des Projektes.

Endprodukt

Das Endprodukt der Projektgruppe AI Börse ist eine grafische Benutzeroberfläche, welche die Daten aus der Datenbank anzeigt und die angeforderten Funktionalitäten passend ausführt. Durch die neuen Herausforderungen und Anforderungen die während des Projekts auftauchten, kam es zu Änderungen der GUI, weswegen sie jetzt anders aussieht als geplant.

Die grafische Benutzeroberfläche hat 2 Darstellungen, diese unterscheiden sich in der Börsenöffnung. In der einen wird die Liste der heutigen Börsenbriefe angezeigt, wenn die Börse offen ist und in der anderen nicht, wenn die Börse geschlossen wurde.

Beide Darstellungen werden in den nächsten Abbildungen (Abbildung 12.11 und 12.12) dargestellt und genau beschrieben.



The screenshot shows the graphical user interface of the AI Börse project. At the top left is the logo of HTL St. Pölten, which consists of a stylized 'HTL' in green and grey, with 'HTL ST. PÖLTEN' written below it in a smaller, bold, grey font. To the right of the logo is the text 'AI BÖRSE' in a red and black sans-serif font. Below the logo and text, there is a message: 'Derzeitiger DAX-Wert: Die Börse ist geschlossen!' followed by the timestamp 'Wed Mar 24 2021 08:26:42 GMT+0100 (Mitteleuropäische Normalzeit)'. Underneath this message is a dropdown menu labeled 'Zeitraum auswählen:' with the option 'Gesamter Zeitraum' selected. The main content area is titled 'Liste der Börsennewsseiten' and contains a table with data for various news websites. The table has columns for 'Börsennewsseite', 'Wahrschein. pos. Prog. 1', 'Wahrschein. pos. Prog. 2', 'Wahrschein. neg. Prog. 1', and 'Wahrschein. neg. Prog. 2'. The data is as follows:

Börsennewsseite	Wahrschein. pos. Prog. 1	Wahrschein. pos. Prog. 2	Wahrschein. neg. Prog. 1	Wahrschein. neg. Prog. 2
www.onvista.de	25%	7%	17%	3%
app.libertex.com	31%	0%	4%	0%
finanzmarktwelt.de	70%	62%	62%	37%
admiralmarkets.com	82%	79%	71%	56%
www.godmode-trader.de	41%	27%	27%	16%
www.boerse-daily.de	44%	15%	48%	21%

Below the table is another section titled 'Liste der heutigen Börsenbriefe'. This section includes a table with columns for 'Börsenbrief', 'Erreichte Vorhersage', 'Bevorzugte nächste Anlaufstelle', 'Wahrscheinlichkeit', 'Alternative Anlaufstelle', and 'Alternative Wahrscheinlichkeit'. A note below the table states 'Die Börse ist geschlossen!'. The text 'Abbildung 12.11: GUI nach Börsenschluss' and 'Quelle: Projektteam AI Börse' is located at the bottom of the screenshot.

Abbildung 12.11: GUI nach Börsenschluss
Quelle: Projektteam AI Börse



Derzeitiger DAX-Wert: 14588

Wed Mar 24 2021 17:12:18 GMT+0100 (Mitteleuropäische Normalzeit)

Zeitraum auswählen:

Gesamter Zeitraum

Liste der Börsennewsseiten

Börsennewsseite	Wahrschein. pos. Prog. 1	Wahrschein. pos. Prog. 2	Wahrschein. neg. Prog. 1	Wahrschein. neg. Prog. 2
www.onvista.de	25%	7%	17%	3%
app.libertex.com	31%	0%	4%	0%
finanzmarktwelt.de	70%	62%	62%	37%
admiralmarkets.com	82%	79%	71%	56%
www.godmode-trader.de	41%	27%	27%	16%
www.boerse-daily.de	44%	15%	48%	21%

Liste der heutigen Börsenbriefe

Börsenbrief	Erreichte Vorhersage	Bevorzugte nächste Anlaufstelle	Wahrscheinlichkeit	Alternative Anlaufstelle	Alternative Wahrscheinlichkeit
https://finanzmarktwelt.de/dax-daily-der-leitinde...	-	14608	70%	14535	62%
https://admiralmarkets.com/de/analysen/dax30-t...	-	14590	82%	14537	71%
https://www.boerse-daily.de/boersen-nachrichten...	-	14600	44%	14500	48%
https://www.onvista.de/news/dax-unschoene-kur...	-	14600	25%	14539	17%
https://www.godmode-trader.de/analyse/dax-tag...	14550	14600	27%	14461	27%
https://www.onvista.de/news/dax-auf-los-gehts-l...	-	14539	17%	14707	25%

Abbildung 12.12: GUI vor Börsenschluss

Quelle: Projektteam AI Börse

Wie man gut erkennen kann, wird, wenn die Börse geschlossen hat, der momentane DAX und die Liste der heutigen Börsenbriefe nicht angezeigt. Das hat den plausiblen Grund, dass es keine Prognosen oder Kurswerte gibt, während die Börse geschlossen ist und somit Anzeigefehler vermieden werden.

Nun wird die Grafik mit der geöffneten Börse genauer beschrieben, da sich ja einiges zur Skizze, die am Anfang des Projekts gezeichnet wurde, verändert hat.

Die markanten Änderungen und bestehenden Elemente kurz zusammengefasst:

- GUI beinhaltet immer noch das Logo der HTL St.Pölten und des Projektteams AI Börse
- Aussehen und Funktion der beiden Tabellen wurde geändert
- Spalten und Position der Tabellen wurden editiert
- Drop-Down-Menü wurde aus der ersten Tabelle entfernt und darüber eingefügt
- Textfeld mit dem momentanen DAX-Wert wurde eingefügt
- Aussehen der momentanen Uhrzeit wurde geändert
- Buttons und Textfelder wurden entfernt (Aufgrund der Tatsache, dass der Crawler die Daten automatisch lädt und deswegen keine manuelle Eingabe benötigt wird)
- Funktion, dass Daten nicht mehr angezeigt werden (nach Börsenschluss)
- Gefärbte Zeilen bei verschiedenen Börsenaussagen

12.4.5 Programmierung der GUI

Die Programmierung der grafischen Benutzeroberfläche wird mittels der Codezeilen aus der Codeigniter-Applikation veranschaulicht und erklärt:

Grafiken:

```
1 div class="container">
2     <div class="row" style="padding-top: 80px; padding-bottom: 50px">
3         <div class="col">
4             
6         </div>
7         <div class="col">
8             
11        </div>
12    </div>
13
```

Listing 12.1: Anzeigen der Grafiken

In diesem Codeausschnitt werden zwei Grafiken mit einer gewissen Höhe und Breite erstellt. Für die Formatierung und Positionierung der Bilder sind die div- und style-Elemente zuständig. Jedes img-Element besitzt eine Source, aus der das Bild geladen

wird, noch dazu besitzt es einen Standard-Text der angezeigt wird, wenn das Bild nicht dargestellt werden kann.

Aktualisierung der GUI-Elemente:

```

1  function load_page(days) {
2      load_dax();
3      load_timestamp();
4      load_table_1(days);
5      load_table_2(days);
6  }

```

Listing 12.2: GUI-Elemente werden aktualisiert

Hier werden Methoden aufgerufen, welche die Daten in den Tabellen und Textfeldern neu befüllen (Aktualisierungsvorgang), nachdem die Funktion „load_page()“ aufgerufen wurde. Diese Methode wird aufgerufen, wenn ein neues Element in der Dropdown-Liste selektiert wird oder die Seite (neu-)geladen wird.

Die aufgerufenen Methoden: load_dax(), load_timestamp(), load_table_1 und load_table_2, werden in den nächsten Listings ausführlich erklärt.

Dropdown-Liste:

```

1 <span>Zeitraum auswählen:</span>
2     <select onchange="load_page(this.value)" class="form-select">
3         <option value="">Gesamter Zeitraum</option>
4         <option value="7">7 Tage</option>
5         <option value="30">30 Tage</option>
6         <option value="90">3 Monate</option>
7         <option value="180">6 Monate</option>
8         <option value="365">1 Jahr</option>
9     </select>

```

Listing 12.3: Anzeigen der Dropdown-Liste

Hier wird die Dropdown-Liste mit ihrem Textfeld „Zeitraum auswählen“ angezeigt. Die folgenden Werte können ausgewählt werden: 7 Tage, 30 Tage, 3 Monate, 6 Monate, 1 Jahr und Gesamter Zeitraum, dabei ist das letzte schon vorselektiert und wird beim Öffnen der Seite bereits ausgewählt, somit werden die entsprechenden Tabellen mit den passenden Daten gefüllt sein.

Beim Wählen eines Elementes der Liste wird die Funktion „onchange()“ aufgerufen, welche die Daten auf der Seite aktualisiert und damit die entsprechenden Werte anzeigen lässt. Dies funktioniert, da das onchange-Event jegliche Veränderung bemerkt und die Funktion aufruft, dabei wird das momentan selektierte Element mitgeschickt und

verarbeitet. Dazu zählen wie im Listing 12.2 zu sehen ist, die Tabellen, der DAX-Wert und die Zeit.

DAX-Wert:

```
1 Derzeitiger DAX-Wert: <span id="dax_value" style="font-size: 130%;"></span>
```

Listing 12.4: DAX-Wert darstellen

Dieser Code veranschaulicht wie der DAX-Wert auf der Webseite angezeigt wird. Dazu gehört das Textfeld „Derzeitiger DAX-Wert:“ und das span-Element, in dem eine ID gesetzt ist. Durch die gesetzte ID kann man mittels Funktionen den Inhalt ändern. Für den Inhalt des span-Elementes wurde eine feste Größe festgelegt.

```
1 function load_dax() {
2     document.getElementById("dax_value").innerHTML = "Wird geladen...";
3     var xmlhttp = new XMLHttpRequest();
4     xmlhttp.onreadystatechange = function() {
5         if (this.readyState == 4 && this.status == 200) {
6             document.getElementById("dax_value").innerHTML = this.
7                 responseText;
8         }
9     };
10    xmlhttp.open("GET", "ci/get_current_dax", true);
11    xmlhttp.send();
}
```

Listing 12.5: AJAX Aufruf zum erhalten des DAX-Wertes

Diese Funktion ruft mittels AJAX eine weitere Funktion (siehe Listing 12.6) mit der Route ci/get_current_dax auf. Die Routen wurden in dem File „Routes.php“ erstellt und somit können ausgewählte Controller angesprochen werden, die nötige Funktionen ausführen.

Nachdem der Aufruf in der Funktion gültig ist (das heißt die If-Abfrage erfüllt hat), wird der übergebene String ausgegeben, ansonsten steht der Text: „Wird geladen...“ im span-Element mit der ID = „dax_value“ solange der Inhalt geladen wird.

```
1 public function get_current_dax() {
2     $dax = file_get_contents("http://10.139.0.36/api/current_dax");
3     echo $dax;
4 }
```

Listing 12.6: Darstellung des aktuellen DAX-Wertes

Die Aufgabe dieser Methode ist es den aktuellen DAX-Wert in eine Variable zu speichern und auszugeben. Den aktuellen DAX-Wert kann man mit der URL

„http://10.139.0.36/api/current_dax“ erreichen. Somit kann man immer auf die aktuellen Werte zugreifen, ohne sie umständlich mit anderen Möglichkeiten auszulesen oder abzufragen.

Timestamp:

```
1 <span id="timestamp"></span>
```

Listing 12.7: Darstellung des aktuellen Timestamps

In diesem Listing wird ein span-Element mit einer gesetzten ID gezeigt. Mithilfe dieser ID kann man darauf zugreifen und den Inhalt des span-Elementes ändern.

```
1 function load_timestamp() {
2     document.getElementById("timestamp").innerHTML = new Date().toString
3 }
```

Listing 12.8: Erhalten der aktuellen Zeit

Die Funktion lädt den neuen Timestamp in das Element mit der ID = „timestamp“. Dies ist durch die Funktion „Date()“ möglich, diese liefert eine formatierte Datumszeichenfolge zurück.

Liste der Börsennewsseiten:

```
1 <h4>Liste der Boersennewsseiten</h4>
2     <table class="table table-hover table-bordered">
3         <thead class="align-middle">
4             <tr>
5                 <th>Boersennewsseite</th>
6                 <th>Wahrschein. pos. Prog. 1</th>
7                 <th>Wahrschein. pos. Prog. 2</th>
8                 <th>Wahrschein. neg. Prog. 1</th>
9                 <th>Wahrschein. neg. Prog. 2</th>
10            </tr>
11        </thead>
12        <tbody id="table_1"></tbody>
13    </table>
```

Listing 12.9: Liste von Börsennewsseiten darstellen

Als allererstes wird der Text „Liste der Börsennewsseiten.“ ausgegeben, dies geschieht mit einem h4-Element was dazu führt, dass der Text fett und etwas größer geschrieben wird.

Nach dem Titel wird die Grundstruktur mit den dazugehörigen Spaltentiteln der Tabelle erstellt. Der Tabellenkörper wird mit einer ID = „table_1“ versehen, somit kann man den Tabellenkörper flexibel ändern.

```

1 function load_table_1(days) {
2     document.getElementById("table_1").innerHTML = "Wird geladen...";
3     var xmlhttp = new XMLHttpRequest();
4     xmlhttp.onreadystatechange = function() {
5         if (this.readyState == 4 && this.status == 200) {
6             document.getElementById("table_1").innerHTML = this.responseText;
7         }
8     };
9     xmlhttp.open("GET", "ci/get_table_1?days=" + days, true);
10    xmlhttp.send();
11}

```

Listing 12.10: AJAX Aufruf für die Börsennewsseiten

Diese Funktion ruft mittels AJAX eine weitere Funktion (siehe Listing 12.11) mit der Route ci/get_table_1 auf. Die Routen wurden in dem File „Routes.php“ erstellt und somit können ausgewählte Controller angesprochen werden, die nötige Funktionen ausführen. In diesem Fall werden beim Aufruf die Tage die in der DropDownList selektiert sind mitgeschickt, damit die richtigen Daten zurückgeschickt werden können.

Es sind beispielsweise 7 Tage in der DropDownList ausgewählt worden, werden diese mitgeschickt, damit die Werte die zurückgeschickt werden, auch im Zeitraum der 7 Tage sind.

Nachdem der Aufruf in der Funktion gültig ist (das heißtt die If-Abfrage erfüllt hat) wird der übergebene String ausgegeben, ansonsten steht der Text: „Wird geladen...“ im span-Element mit der ID = „table_1“ solange der Inhalt geladen wird.

```

1 public function get_table_1() {
2     $days = $_REQUEST["days"];
3     if ($days == "") {
4         $json_data = file_get_contents("http://10.139.0.36/api/auswertung2");
5     }
6     else {
7         $json_data = file_get_contents("http://10.139.0.36/api/auswertung2/" .
8             $days);
9     }
10    $json_data = '{"status":"success", "data":' . $json_data . '}';
11    $json_data = str_replace("\n", " ", $json_data);
12    $json_data = str_replace("\r", "\n", $json_data);
13
14    $response_data = json_decode($json_data);
15    $t1 = $response_data->data;

```

```

15 foreach ($t1 as $ausw) {
16   echo "<tr>";
17   echo "<td>" . $ausw->Boersennewsseite . "</td>";
18   echo "<td>" . $ausw->Wahrsch_pos_Prog_1 . "</td>";
19   echo "<td>" . $ausw->Wahrsch_pos_Prog_2 . "</td>";
20   echo "<td>" . $ausw->Wahrsch_neg_Prog_1 . "</td>";
21   echo "<td>" . $ausw->Wahrsch_neg_Prog_2 . "</td>";
22   echo "</tr>";
23 }
24 }
```

Listing 12.11: Neue Daten für die Börsennewsseiten aufrufen

Die 2. Zeile des Listings zeigt die Speicherung des übergebenen Wertes in eine Variable in der Methode. Nach dieser Speicherung wird indirekt abgeprüft ob der übergebene Wert leer war oder nicht, wenn das Erstere der Fall ist werden die Daten für den ganzen Zeitraum aufgerufen, da in der Dropdown-Liste der gesamte Zeitraum die einzige Auswahl ohne Wert war (siehe Listing 12.3). Wenn ein Wert übergeben wurde, dann wird dieser im Aufruf mitgeschickt und somit bekommt man die passenden Daten zurückgeschickt.

Der Aufruf geschieht über die URL „<http://10.139.0.36/api/auswertung2>“ , wenn ein Wert vorhanden ist wird dieser auch mit angehängt und geschickt. Als Rückgabewert bekommt man einen JSON-String der formatiert werden muss, damit er verwendet werden kann. Da dieser übergebene String keinen Header besitzt und man ihn ohne Header nicht decoden kann, wird einer gesetzt. Ein weiteres Problem sind die Whitespaces und Punkte, weil diese Satzzeichen beim encoden des Strings zu einer Fehlermeldung führen.

Um das Problem zu umgehen wurden zuerst alle Punkte durch Whitespaces ersetzt und danach wurden alle Whitespaces mit Unterstrichen ersetzt, dies löste die Probleme. Nachdem der String korrekt formatiert wurde, ist die Funktion zum encoden des Strings aufgerufen worden. Dieser liefert eine JSON-codierte Zeichenfolge aus, wenn diese ohne Fehler verlief und bei Misserfolg lieferte die Funktion false.

Zum Schluss werden in einer Schleife alle passenden Daten in Form von Tabellenzeilen erstellt und zurück geschickt, damit man diese dann in der grafischen Oberfläche anzeigen lassen kann.

Liste der heutigen Börsenbriefe:

```

1 <h4>Liste der heutigen Boersenbriefe</h4>
2   <table class="table table-hover table-bordered">
3     <thead class="align-middle">
4       <tr>
5         <th style="width: 30%;">Boersenbrief</th>
```

```

6   <th>Erreichte Vorhersage</th>
7   <th>Bevorzugte naechste Anlaufstelle</th>
8   <th>Wahrscheinlichkeit</th>
9   <th>Alternative Anlaufstelle</th>
10  <th>Alternative Wahrscheinlichkeit</th>
11  </tr>
12  </thead>
13  <tbody id="table_2"></tbody>
14  </table>

```

Listing 12.12: Liste von den heutigen Börsenbriefe darstellen

Als allererstes wird der Text „Liste der heutigen Börsenbriefe“ ausgegeben, dies geschieht mit einem h4-Element was dazu führt, dass der Text fett und etwas größer geschrieben wird.

Nach dem Titel wird die Grundstruktur der Tabelle mit den dazugehörigen Spaltentiteln erstellt. Der Tabellenkörper wird mit einer ID = „table_2“ versehen, somit kann man den Tabellenkörper flexibel ändern.

```

1 function load_table_2(days) {
2     document.getElementById("table_2").innerHTML = "Wird geladen...";
3     var xmlhttp = new XMLHttpRequest();
4     xmlhttp.onreadystatechange = function() {
5         if (this.readyState == 4 && this.status == 200) {
6             document.getElementById("table_2").innerHTML = this.responseText;
7         }
8     };
9     xmlhttp.open("GET", "ci/get_table_2?days=" + days, true);
10    xmlhttp.send();
11 }

```

Listing 12.13: AJAX Aufruf für die heutigen Börsenbriefe

Diese Funktion ruft mittels AJAX eine weitere Funktion (siehe Listing 12.14) mit der Route ci/get_table_1 auf. Die Routen wurden in dem File „Routes.php“ erstellt und somit können ausgewählte Controller angesprochen werden, die nötige Funktionen ausführen. In diesem Fall werden beim Aufruf die Tage die in der DropDown-Liste selektiert sind mitgeschickt, damit die richtigen Daten zurückgeschickt werden können.

Nachdem der Aufruf in der Funktion gültig ist (das heißt die If-Abfrage erfüllt hat), wird der übergebene String ausgegeben, ansonsten steht der Text: „Wird geladen...“ im span-Element mit der ID = „table_2“ solange der Inhalt geladen wird.

```

1 public function get_table_2() {
2     $days = $_REQUEST["days"];

```

```

3 if ($days == "") {
4     $json_data = file_get_contents("http://10.139.0.36/api/auswertung");
5 }
6 else {
7     $json_data = file_get_contents("http://10.139.0.36/api/auswertung/" .
8         $days);
9 }
10 $json_data = '{"status":"success", "data":' . $json_data . '}';
11 $json_data = str_replace(".", " ", $json_data);
12 $json_data = str_replace(" ", "_", $json_data);

13 $response_data = json_decode($json_data);
14 $t2 = $response_data->data;
15 foreach ($t2 as $ausw) {
16     if ($ausw->Boersennewsseite == "-") {
17         $style = "";
18         echo "Die Börse ist geschlossen!";
19         return;
20     }
21     elseif ($ausw->Bevorzugte_naechste_Anlaufstelle != "-" && $ausw->
22         Bevorzugte_naechste_Anlaufstelle > $ausw->Alternative_Anlaufstelle
23         || $ausw->Erreichte_Vorhersage > $ausw->Alternative_Anlaufstelle)
24     {
25         $style = "background-color: #DDFFDD;";
26     }
27     else {
28         $style = "background-color: #FFDDDD;";
29     }

30     echo "<tr style=\"" . $style . "\">>";
31     echo "<td style=\"white-space: nowrap; overflow: hidden; text-
32         overflow: ellipsis; width: 30%;\"><a href=\"" . $ausw->
33         Boersennewsseite . "\" target=\"_blank\">" . $ausw->
34         Boersennewsseite . "</a></td>";
35     echo "<td>" . $ausw->Erreichte_Vorhersage . "</td>";
36     echo "<td>" . $ausw->Bevorzugte_naechste_Anlaufstelle . "</td>";
37     echo "<td>" . $ausw->Wahrscheinlichkeit . "</td>";
38     echo "<td>" . $ausw->Alternative_Anlaufstelle . "</td>";
39     echo "<td>" . $ausw->Alt_Wahrscheinlichkeit . "</td>";
40     echo "</tr>";
41 }
42 }

```

Listing 12.14: Neue Daten für die heutigen Börsenbriefe aufrufen

Die Funktion regelt das Aufrufen der neuen Daten, die in die Tabelle mit den heutigen Börsenbriefen eingefügt werden. Als erstes wird der übergebene Wert in eine Variable gespeichert, damit sie für die Abfragen genutzt werden kann.

Nach dieser Speicherung wird indirekt abgeprüft, ob der übergebene Wert leer war

oder nicht, wenn das Erstere der Fall ist, werden die Daten für den ganzen Zeitraum aufgerufen, da in der Dropdown-Liste der Gesamte Zeitraum die einzige Auswahl ohne Wert war (siehe Listing 12.3). Wenn ein Wert übergeben wurde, dann wird dieser im Aufruf mitgeschickt und somit bekommt man die passenden Daten zurückgeschickt.

Der Aufruf geschieht über die URL „<http://10.139.0.36/api/auswertung>“ , wenn ein Wert vorhanden ist wird dieser auch mit angehängt und geschickt. Als Rückgabewert bekommt man einen JSON-String der formatiert werden muss, damit er verwendet werden kann. Da dieser übergebene String keinen Header besitzt und man ihn nicht ohne Header decoden kann, wird einer gesetzt. Ein weiteres Problem sind wieder die Whitespaces und Punkte, weil diese Satzzeichen beim encoden des Strings zu einer Fehlermeldung führen.

Um das Problem zu umgehen wurden wie bei der Funktion „function_table_1“ zuerst alle Punkte durch Whitespaces ersetzt und danach wurden alle Whitespaces mit Unterstrichen ersetzt, dies löste die Probleme. Nachdem der String korrekt formatiert wurde, ist die Funktion zum encoden des Strings aufgerufen worden. Dieser liefert eine JSON-codierte Zeichenfolge aus, wenn diese ohne Fehler verlief und bei Misserfolg liefert die Funktion false.

Nachdem die neuen aufgerufenen Daten gespeichert wurden wird im Schleifendurchgang überprüft ob diese leer sind, dies kann passieren, wenn die Börse geschlossen hat (Es gibt keine Börsenbriefe, wenn die Börse geschlossen ist). Wenn der Fall aufgetreten ist, dass die Börse geschlossen ist, wird der Text „Die Boerse ist geschlossen!“ ausgegeben. Im anderen Fall wäre die Variable mit Daten gefüllt und kann dargestellt werden, bevor dies jedoch geschieht wird überprüft ob die bevorzugte nächste Anlaufstelle höher ist als die alternative Anlaufstelle oder ob die erreichte Vorhersage größer als die alternative Anlaufstelle ist. Wenn die If-Abfrage zutrifft, dann wird die Zeile grün eingefärbt, ansonsten wird sie rot eingefärbt.

Im Anschluss werden die neuen Daten in Form von Tabellenzeilen erstellt und zurück geschickt, damit diese dann in der GUI abgebildet werden können.

Die 2. Zeile des Listings zeigt die Speicherung des übergebenen Wertes in eine Variable in der Methode. Nach dieser Speicherung wird indirekt abgeprüft, ob der übergebene Wert leer war oder nicht, wenn das Erstere der Fall ist, werden die Daten für den ganzen Zeitraum aufgerufen, da in der Dropdown-Liste der Gesamte Zeitraum die einzige Auswahl ohne Wert war (siehe Listing 12.3). Wenn ein Wert übergeben wurde wird dieser im Aufruf mitgeschickt und somit bekommt man die passenden Daten zurückgeschickt.

12.4.6 Fazit

Nach so manchen Änderungen und zusätzlichen Wünschen entstand ein gutes und zufriedenstellendes Endprodukt. Wie man schon aus den vorigen Kapiteln herauslesen konnte, brachte dieser Teil des Projektes eine Menge Arbeit mit sich. Von der Planung bis zur Implementation der grafischen Oberfläche gab es Hoch- und Tiefpunkte, jedoch kam zum Schluss ein sehr zufriedenstellendes und verwertbares Endprodukt heraus. Dennoch sollte man das Endprodukt des Projektes AI-Börse nicht direkt in andere Projekte hinein implementieren wenn man nicht zuvor Änderungen unternommen hat, um es auf das spezifische Projekt abzustimmen.

Die Ergebnisse und Hindernisse die im Laufe der Entstehung der GUI entstanden, wurden in diesem Abschnitt protokolliert. Besonders die Erfahrungen die somit niedergeschrieben wurden sind beispielsweise für andere, zukünftige Projekte sehr wertvoll. Beispielsweise wird in der Implementation gezeigt, wie durch Aktionen in einem GUI-Element zum Beispiel eine DropDownList, die Tabellen, der DAX-Wert und die aktuelle Zeit (Timestamp) geändert werden, dies betitelt die Projektgruppe als Aktualisierungsvorgang. Da die Funktionalitäten der GUI selbst geschrieben wurden könnte es sein, dass ähnliche Methoden im Internet nur schwer auffindbar sind.

Das Planen der GUI war auch etwas Neues für das erste Projekt der vier jungen Männer, deswegen ist das Praxisbeispiel besonders für neue Projekte mit nicht so erfahrenen Mitgliedern äußerst ansprechend. Diese können aus einfachen Fehlern des Projektes AI-Börse lernen und dadurch ihr Projekt schneller voranbringen oder qualitativ steigern.

Abbildungsverzeichnis

2.1	Sternstruktur	8
2.2	Lineare Hyperlinkstruktur	9
2.3	Netzstruktur	9
2.4	Baumstruktur	10
2.5	Title-Tag des RYTE Wiki	14
2.6	Description-Tag des RYTE Wiki	14
2.7	Heading-Tags h1 - h6	17
3.1	Architektur eines Crawlers	29
3.2	Arbeitsprozess eines Crawlers	30
3.3	Breath-First-Search vs Depth-First-Search	32
3.4	Architektur eines Focused Crawlers	33
3.5	Architektur eines Distributed Crawlers	34
3.6	User-Interface von idealo.at	35
4.1	Arbeitsprozess Link- & Contentcrawler	53
4.2	Arbeitsprozess Libertexcrawler	54
5.1	Optimierung durch Parallelisierung	68

5.2	Geschwindigkeitsgewinn im Bezug zum Amdahl's Law	68
6.1	Komponenten von Hadoop	72
6.2	Die Architektur von HDFS	73
6.3	Die Architektur von YARN	75
6.4	Ablauf von MapReduce	76
6.5	Apache Hadoop Ecosystem	78
6.6	Apache Spark Verbreitung.	84
6.7	Einbinden von Apache Spark in Hadoop.	86
6.8	Hauptbestandteile von Spark	87
6.9	Spark RDD	88
6.10	Veranschaulichung von Spark Streaming	90
6.11	Logistic Regression Pipeline	95
6.12	Logistic Regression Pipelinemodell	95
6.13	Pythonic Graph 2013	101
6.14	Pythonic Graph 2020	101
6.15	Spark als ETL Engine	105
7.1	Unterschiedlichen Version von Apache Spark	111
7.2	Hash einer Apache Spark Version	112
7.3	Anzeige der HOME Variablen	113
7.4	Anzeige der Path Variable	113
9.1	Syntax von „The dog chased the cat.“	130
9.2	Funktions-Argument-Struktur von „Philip ist männlich“	132

9.3	Diskursanalyse bei der Textgenerierung	138
9.4	Question Answering des Google Assistant	139
9.5	One-Hot Encoding	146
9.6	Vektordarstellungen mithilfe von Word2Vec	148
9.7	Logistische Funktion	149
9.8	Veranschaulichung des <i>Threshold Values</i>	150
9.9	Ausgleichsgerade (rot) der geg. Daten (blau)	151
9.10	Visualisierung der Vektorformen	153
9.11	Veranschaulichung des k-Means-Algorithmus	156
9.12	Ellbogenmethode	158
9.13	10 wichtigsten Wörter pro Kategorie	158
9.14	Das biologische Neuron	160
9.15	Aufbau eines künstlichen Neurons	161
9.16	Grundsätzlicher Aufbau eines künstlichen neuronalen Netzes	162
9.17	Recurrent Neural Network	163
9.18	Veranschaulichung einer LSTM-Zelle	165
9.19	Convolutional Neural Network	166
10.1	Erfasste Prognosewerte in Tabellenform	169
10.2	Typische spaCy Pipeline	170
11.1	Schnittpunkt von BigData / ML und Systems	188
11.2	Ablauf der Daten	191
11.3	Ablauf der Daten	193
11.4	Darstellung von CRUD	196

11.5	Logo AI Börse	200
11.6	Verarbeitung der Daten	201
11.7	Sankey Diagram	203
12.1	Datenpyramide	206
12.2	Arten von Daten	207
12.3	Ausschnitt von Daten von der AI Börse	210
12.4	Trefferquote von Prognosen einer Börsenseite in Prozent	213
12.5	Durchschnittlicher Gewinn in den einzelnen Börsenmonaten	214
12.6	Durchschnittliche Trefferquote von richtigen Prognosen	216
12.7	Aufbau einer GUI	219
12.8	Designsprachen	225
12.9	Skizze der GUI	228
12.10	Funktionsweise von CodeIgniter	231
12.11	GUI nach Börsenschluss	233
12.12	GUI vor Börsenschluss	234

Listings

2.1	Hyperlink auf einer Webseite einbinden	5
2.2	Bild mit Hyperlink hinterlegen	6
2.3	Verweis auf eine Unterseite	6
2.4	Outbound Link im HTML-Body	7
2.5	Hyperlink um ein PDF zu öffnen	7
2.6	Hyperlink zum Öffnen eines E-Mail-Programmes	8
2.7	Canonical-URL mit Link-Tag erzeugen	16
2.8	Erzeugen einer Canonical-URL per HTTP-Header	16
2.9	Heading-Tags im HTML-Body	17
2.10	Beispielaufbau einer XML-Sitemap	19
3.1	Scrapy Beispiel	47
3.2	Scrapy Beispieldatei	48
3.3	Jaunt Beispiel	49
3.4	Jaunt Beispieldatei	49
3.5	Goutte Beispiel	50
3.6	Goutte Beispieldatei	50
4.1	Linkcrawler	55
4.2	outputLinks.json	56

4.3	Contentcrawler	57
4.5	Ausgabe Contentcrawler	58
4.6	Libertexcrawler	59
4.8	Ausgabe Libertexcrawler	61
4.9	CrawlingScript.sh	62
6.1	Example Pipeline	97
6.2	Example Pipeline Output1	98
7.1	Überprüfung der Java Version auf dem Server	110
7.2	Verhashen von Apache Spark	112
7.3	Festlegen von Rechten von \tmp	113
7.4	PySpark starten	114
7.5	Einbindung von Pipelines	115
7.6	Output praxis Pipeline1	117
7.7	Output praxis Pipeline2	118
7.8	Installation Spark auf Linux	119
7.9	Benutzung PySpark DataFrames	120
7.10	PySpark DataFrames Output1	121
7.11	PySpark DataFrames Output2	121
7.12	PySpark DataFrames Output3	122
7.13	PySpark DataFrames Output4	122
7.14	PySpark DataFrames Output5	122
7.15	PySpark Datenbankzugriff	123
7.16	db_properties	123

7.17	Import Koalas	123
9.1	Syntaxanalyse mithilfe von spaCy	130
9.2	Tokenisierung mithilfe von spaCy	135
9.3	Part-of-Speech-Tagging mithilfe von spaCy	136
9.4	Maschinelle Übersetzung mithilfe von textblob	140
9.5	Logistische Regression des Datensatzes	153
10.1	Format der Trainingsdaten	171
10.2	Trainingsdaten für die Named Entity Recognition	171
10.3	Training des NER-Modells	172
10.4	Speichern des NER-Modells	173
10.5	Laden des NER-Modells	173
10.6	Testen des NER-Modells	173
10.7	Methoden der Datenbankschnittstelle	175
10.8	Aufteilung der Texte in einzelne Sätze	176
10.9	Übersetzung der Sätze	177
10.10	analyze_sentiment	177
10.11	TextBlob	178
10.12	TextBlobDE	178
10.13	NLTK VADER	178
10.14	Definition der Stichwörter	182
10.15	Vorbereitung der Texte	182
12.1	Anzeigen der Grafiken	235
12.2	GUI-Elemente werden aktualisiert	236

12.3 Anzeigen der Dropdown-Liste	236
12.4 DAX-Wert darstellen	237
12.5 AJAX Aufruf zum erhalten des DAX-Wertes	237
12.6 Darstellung des aktuellen DAX-Wertes	237
12.7 Darstellung des aktuellen Timestamps	238
12.8 Erhalten der aktuellen Zeit	238
12.9 Liste von Börsennewsseiten darstellen	238
12.10 AJAX Aufruf für die Börsennewsseiten	239
12.11 Neue Daten für die Börsennewsseiten aufrufen	239
12.12 Liste von den heutigen Börsenbriefe darstellen	240
12.13 AJAX Aufruf für die heutigen Börsenbriefe	241
12.14 Neue Daten für die heutigen Börsenbriefe aufrufen	241

Literaturverzeichnis

- [NP:Web01] More-Fire: Crawling und Indexierung von Webseiten: Theorie und Praxis. Online im Internet: URL: <https://www.more-fire.com/blog/crawling-und-indexierung-von-webseiten-theorie-und-praxis/>, 27. Oktober 2020
- [NP:Web02] Xovi: Was bedeutet Indexierung? Online im Internet: URL: <https://www.xovi.de/was-bedeutet-indexierung/>, 28. Oktober 2020
- [NP:Web03] Ryte Wiki: Crawl Budget. Online im Internet: URL: https://de.ryte.com/wiki/Crawl_Budget, 28. Oktober 2020
- [NP:Web04] Ryte Wiki: PageRank. Online im Internet: URL: <https://de.ryte.com/wiki/PageRank>, 28. Oktober 2020
- [NP:Web05] Ryte Wiki: Random Surfer Model. Online im Internet: URL: https://de.ryte.com/wiki/Random_Surfer_Model, 28. Oktober 2020
- [NP:Web06] HTML-Seminar: HTML-Links. Online im Internet: URL: <https://www.html-seminar.de/html-links.htm>, 30. Oktober 2020
- [NP:Web07] HTML-Seminar: Interne Links. Online im Internet: URL: <https://www.html-seminar.de/interne-links.htm>, 30. Oktober 2020
- [NP:Web08] Advidera: Hyperlinks. Online im Internet: URL: <https://www.advidera.com/glossar/hyperlink/>, 30. Oktober 2020
- [NP:Web09] IONOS: Hyperlinks. Online im Internet: URL: <https://www.ionos.at/digitalguide/websites/web-entwicklung/hyperlink/>, 5. November 2020
- [NP:Web10] Eology: Hyperlinks. Online im Internet: URL: <https://www.eology.de/wiki/hyperlink>, 5. November 2020
- [NP:Web11] Ryte Wiki: Suchmaschinenoptimierung. Online im Internet: URL: <https://de.ryte.com/wiki/Suchmaschinenoptimierung>, 9. November 2020
- [NP:Web12] Ryte Wiki: Onpage-Optimierung. Online im Internet: URL: https://de.ryte.com/wiki/OnPage_Optimierung, 9. November 2020

- [NP:Web13] Vioma: Onpage-Optimierung. Online im Internet: URL: <https://www.vioma.de/de/wiki/online-marketing/seo/onpage-optimierung/>, 9. November 2020
- [NP:Web14] RYTE Wiki: Canonical-Tag. Online im Internet: URL: https://de.ryte.com/wiki/Canonical_Tag, 13. November 2020
- [NP:Web15] Google Developers Docs: Consolidate Duplicate URLs. Online im Internet: URL: <https://developers.google.com/search/docs/advanced/crawling/consolidate-duplicate-urls>, 13. November 2020
- [NP:Web16] RYTE Wiki: Google Mobile-Friendly Update. Online im Internet: URL: https://de.ryte.com/wiki/Google_Mobile-Friendly_Update, 21. November 2020
- [NP:Web17] RYTE Wiki: Offpage-Optimierung. Online im Internet: URL: https://de.ryte.com/wiki/OffPage_Optimierung, 22. November 2020
- [NP:Web999] Bacher, Gerald: Offpage-Optimierung. Online im Internet: URL: <https://www.evergreenmedia.at/glossar/offpage-optimierung/>, 24. November 2020
- [NP:Web18] Audisto: URL-Normalisierung. Online im Internet: URL: <https://audisto.com/help/crawler/settings/url-normalization/>, 29. Dezember 2020
- [NP:Web19] QAZ Wiki: URI-Normalisierung. Online im Internet: URL: https://de.qaz.wiki/wiki/URI_normalization, 29. Dezember 2020
- [NP:Web20] Key-shortcut: Zeichentabelle, Prozent-codierte Zeichen. Online im Internet: URL: <https://www.key-shortcut.com/zeichentabellen/ascii-url-kodierung>, 30. Dezember 2020
- [NP:Web21] Google Developers Docs: Googlebot. Online im Internet: URL: <https://developers.google.com/search/docs/advanced/crawling/googlebot>, 2. Jänner 2021
- [NP:Web22] Techopedia: Spider-Trap. Online im Internet: URL: <https://www.techopedia.com/definition/5197/spider-trap>, 2. Jänner 2021
- [NP:Web23] Wiki SelfHTML: Robots.txt. Online im Internet: URL: <https://wiki.selfhtml.org/wiki/Grundlagen/Robots.txt>, 4. Jänner 2021
- [NP:Web24] RYTE Wiki: Robots.txt. Online im Internet: URL: <https://de.ryte.com/wiki/Robots.txt>, 4. Jänner 2021
- [NP:Web25] Mindshape: Robots.txt. Online im Internet: URL: <https://www.mindshape.de/kompetenzen/inbound-marketing/suchmaschinenoptimierung-seo/robotstxt.html>, 5. Jänner 2021

- [NP:Web26] SEO-Südwest: Spider-Trap. Online im Internet: URL: <https://www.seo-suedwest.de/seo-wissen/seo-glossar/spider-trap.html>, 6. Jänner 2021
- [NP:Web27] Sitemaps: Sitemap.xml. Online im Internet: URL: <https://www.sitemaps.org/de/protocol.html>, 7. Jänner 2021
- [NP:Web28] Marketingtracer: Crawler-Traps. Online im Internet: URL: <https://www.marketingtracer.com/seo/crawler-traps>, 30. Jänner 2021
- [NP:Web29] Portent: Field guide to spider traps. Online im Internet: URL: <https://www.portent.com/blog/seo/field-guide-to-spider-traps-an-seo-companion.htm>, 31. Jänner 2021
- [NP:Web30] Scrapy: Scrapy Documentation. Online im Internet: URL: <https://docs.scrapy.org/en/latest/index.html>, 2. Februar 2021
- [NP:Web31] Softkraft: How to web scrape with python. Online im Internet: URL: <https://www.softkraft.co/how-to-web-scrape-with-python/>, 2. Februar 2021
- [NP:Web32] Datahut: Scraping Amazon reviews. Online im Internet: URL: <https://blog.datahut.co/scraping-amazon-reviews-python-scrapy/>, 2. Februar 2021
- [NP:Web33] Open4Tech: BFS vs DFS. Online im Internet: URL: <https://open4tech.com/bfs-vs-dfs/>, 3. Februar 2021
- [NP:Web34] Vabelhavt: Wie funktionieren Web Crawler?. Online im Internet: URL: <https://www.vabelhavt.at/crawler/>, 4. Februar 2021
- [NP:Web35] Goutte: Github FriendsOfPHP Goutte. Online im Internet: URL: <https://github.com/FriendsOfPHP/Goutte>, 5. Februar 2021
- [NP:Web36] Symfony: Components BrowserKit. Online im Internet: URL: https://symfony.com/doc/current/components/browser_kit.html, 5. Februar 2021
- [NP:Web37] Symfony: Components DomCrawler. Online im Internet: URL: https://symfony.com/doc/current/components/dom_crawler.html, 5. Februar 2021
- [NP:Web38] Symfony: Components CssSelector. Online im Internet: URL: https://symfony.com/doc/current/components/css_selector.html, 5. Februar 2021
- [NP:Web39] Symfony: Components HTTP Client. Online im Internet: URL: https://symfony.com/doc/current/http_client.html, 5. Februar 2021
- [NP:Web40] Medium: PHP Webscraping in Goutte. Online im Internet: URL: <https://medium.com/@matvey.nikon/php-web-scraping-in-goutte-e964408d6849>, 6. Februar 2021

- [NP:Web41] Advidera: Bingbot. Online im Internet: URL: <https://www.advidera.com/glossar/bingbot/>, 9. Februar 2021
- [NP:Web42] Bing: Webmasters Help, Which crawlers does bing use. Online im Internet: URL: <https://www.bing.com/webmasters/help/which-crawlers-does-bing-use-8c184ec0>, 10. Februar 2021
- [NP:Web43] DuckDuckGo: DuckDuckGo Help pages. Online im Internet: URL: <https://help.duckduckgo.com/duckduckgo-help-pages/>, 10. Februar 2021
- [NP:Web44] Jaunt: Jaunt API Documentation. Online im Internet: URL: <https://jaunt-api.com/>, 11. Februar 2021
- [NP:Web45] Jauntium: Jauntium Documentation. Online im Internet: URL: <https://jauntium.com/>, 11. Februar 2021
- [NP:Article01] S.AMUDHA, B.SC., M.SC., M.PHIL.: Web crawler for mining web data. International Research Journal of Engineering and Technology (IRJET), e-ISSN: 2395-0056, 2017, pp. 128-136
- [NP:Article02] Drescher, Ringo: Entwicklung eines fokussierten Crawlers für Internetforen. Technische Universität Dresden, 2010, pp. 13-17
- [NP:Article03] Manning, Christopher D.; Raghavan, Prabhakar; Schütze, Hinrich: An Introduction to Information Retrieval. Cambridge University Press, 2009, pp. 443-459
- [NP:Article04] Herta, Christian: Einführung in Webspider. Vorlesung Suchtechnologie und Information Retrieval, 2009, pp. 1-31
- [NP:Article05] Mini, Singh Ahuja; Jatinder, Singh Bal; Varnica: Web Crawler: Extracting the Web Data. International Journal of Computer Trends and Technology (IJCTT) Volume 13 Number 3, 2014, pp. 132-137
- [NP:Article06] Linxuan, Yu; Yeli, Li; Qingtao, Zeng; Yanxiong, Sun; Yuning, Bian; Wei, He: Summary of web crawler technology research. Journal of Physics: Conference Series 1449 012036, 2020, pp. 1-7
- [NP:Article07] Srinivasan, Padmini; Mitchell, Joyce; Bodenreider, Olivier; Pant, Gautam; Menczer, Filippo: Web Crawling Agents for Retrieving Biomedical Information, 2002, pp. 1-8
- [NP:Article08] Risha, Gaur; Dilip kumar, Sharma: Focused Crawling with Ontology using Semi-Automatic Tagging for Relevancy. Seventh International Conference on Contemporary Computing (IC3), 2014, pp. 1-7

- [NP:Article09] Imhof, Lucas: Information Retrieval für eine Web Suchmaschine mithilfe von neuronalen Netzen und klassischen Methoden. Hochschule Merseburg, 2019, pp. 1-27
- [LR:Web01] Fujitsu: Informationen zu Storage-Cluster: Online im Internet: URL: <https://www.fujitsu.com/at/products/computing/storage/disk/eternus-dx/storage-cluster/>, 31. Oktober 2020
- [LR:Web02] IT-Administrator: Grundlagen zu Cluster: Online im Internet: URL: https://www.it-administrator.de/themen/server_client/grundlagen/172792.html, 31. Oktober 2020
- [LR:Web03] Rene Büst: Was ist Cluster Computing: Online im Internet: URL: <https://de.cloudflight.io/presse/was-ist-cluster-computing-15/>, 31. Oktober 2020
- [LR:Web04] Archiv von Apache Spark Versionen: Online im Internet: URL: <https://archive.apache.org/dist/spark/>, 31. Dezember 2020
- [LR:Web05] Maziyar Panahi: Stackoverflow Beitrag zur Installation von PySpark: Online im Internet: URL: <https://stackoverflow.com/questions/65054072/spark-nlp-pretrained-model-not-loading-in-windows>, 1. Jänner 2021
- [LR:Web06] Goran Jevtic: Anleitung für die Installation von PySpark: Online im Internet: URL: <https://phoenixnap.com/kb/install-spark-on-windows-10>, 1. Jänner 2021
- [LR:Web07] Beitrag zu Apache ZooKeeper: Online im Internet: URL: <https://www.edureka.co/community/1106/what-zookeeper-what-the-purpose-zookeeper-hadoop-ecosystem>, 4. Jänner 2021
- [LR:Web08] Liquan Pei: Dokumentation von Apache Hadoop: Online im Internet: URL: <https://cwiki.apache.org/confluence/display/Hive/Home>, 6. Jänner 2021
- [LR:Web09] Ian Cook: Apache Hive LanguageManual: Online im Internet: URL: <https://cwiki.apache.org/confluence/display/Hive/LanguageManual>, 6. Jänner 2021
- [LR:Web10] Karen Coppage: Apache Hive GettingStarted: Online im Internet: URL: <https://cwiki.apache.org/confluence/display/Hive/GettingStarted>, 6. Jänner 2021
- [LR:Web11] Intellipaat: What is Hadoop Ecosystem?: Online im Internet: URL: <https://intellipaat.com/blog/tutorial/hadoop-tutorial/hadoop-ecosystem/#Apache-Storm>, 9. Jänner 2020
- [LR:Web12] Matei Zaharia: Databricks Eintrag zu Apache Spark: Online im Internet: URL: <https://databricks.com/spark/about>, 1. Februar 2021
- [LR:Web13] Denny Lee und Jules Damji: Apache Spark Key Terms, Explained: Online im Internet: URL: <https://databricks.com/blog/2016/06/22/apache-spark-key-terms-explained.html>, 3. Februar 2021

- [LR:Web14] Databricks Beitrag zu Apache MLlib: Online im Internet: URL: <https://databricks.com/glossary/what-is-machine-learning-library>, 3. Februar 2021
- [LR:Web15] Offizieller Guid zu Apache Spark MLlib: Online im Internet: URL: <https://spark.apache.org/docs/latest/ml-guide.html>, 5. Februar 2021
- [LR:Web16] JohnSnow: List of Pretrained Pipelines: Online im Internet: URL: <https://nlp.johnsnowlabs.com/docs/en/pipelines>, 12. Februar 2021
- [LR:Web17] Databricks: Use Apache Spark MLlib on Databricks: Online im Internet: URL: <https://docs.databricks.com/applications/machine-learning/train-model/mllib/index.html>, 6. März 2021
- [LR:Web18] KnowledgeHut: Apache Spark Pros and Cons: Online im Internet: URL: <https://www.knowledgehut.com/blog/big-data/apache-spark-advantages-disadvantages>, 13. März 2021
- [LR:Web19] GoogleCloud:BigQuery under the hood: Online im Internet: URL: <https://cloud.google.com/blog/products/bigquery/bigquery-under-the-hood>, 14. März 2021
- [LR:Web20] IntelliPaat:What is Google's Dremel? How is it different from Mapreduce?: Online im Internet: URL: <https://intellipaat.com/community/1879/what-is-googles-dremel-how-is-it-different-from-mapreduce>, 14. März 2021
- [LR:Web21] Google Research: Dremel: Interactive Analysis of Web-Scale Datasets: Online im Internet: URL: <https://research.google/pubs/pub36632/>, 14. März 2021
- [LR:Web22] RedHat: What is a Kubernetes cluster?: Online im Internet: URL: <https://www.redhat.com/en/topics/containers/what-is-a-kubernetes-cluster>, 14. März 2021
- [LR:Web23] ResearchGate: Graph demonstrating Amdahl's Law: Online im Internet: URL: https://www.researchgate.net/figure/Graph-demonstrating-Amdahls-Law_fig2_315696585, 21. März 2021
- [LR:Web24] SparkByExamples: PySpark - Create DataFrame with Examples: Online im Internet: URL: <https://sparkbyexamples.com/pyspark/different-ways-to-create-dataframe-in-pyspark/>, 21. März 2021
- [LR:Web25] towards data science: The Most Complete Guide to pySpark DataFrames: Online im Internet: URL: <https://towardsdatascience.com/the-most-complete-guide-to-pyspark-dataframes-2702c343b2e8#4d56>, 21. März 2021
- [LR:Web26] DataBrick: Community notebook: Online im Internet: URL: <https://community.cloud.databricks.com/>, 21. März 2021

- [LR:Web27] Virtual Workshop: Parallel Programming Concepts and High-Performance Computing: Online im Internet: URL: <https://cvw.cac.cornell.edu/parallel/amdahl>, 23. März 2021
- [LR:Web28] Ionos: Apache Hadoop: distributed storage architecture for data quantities: Online im Internet: URL: <https://www.ionos.co.uk/digitalguide/server/know-how/apache-hadoop-the-framework-for-big-data/>, 23. März 2021
- [LR:Article01] Thomas Joos: Was ist ein Cluster?: Online im Internet: URL: <https://www.datacenter-insider.de/was-ist-ein-cluster-a-5715/>, 31. Oktober 2020
- [LR:Article02] Laurenz Wuttke: Übersicht zu Apache Hadoop: Online im Internet: URL: <https://datasolut.com/apache-hadoop-einfuehrung/>, 1. Jänner 2021
- [LR:Article03] Benjamin Aunkofer: Anleitung zu MapReduce in Hadoop: Online im Internet: URL: <https://data-science-blog.com/blog/201/03/26/distributed-computing-mapreduce-algorithmus/>, 3. Jänner 2021
- [LR:Article04] Data Flair: Apache HBase Tutorial: Online im Internet: URL: <https://data-flair.training/blogs/apache-hbase-tutorial/>, 6. Jänner 2021
- [LR:Article05] Anshul Aggarwal: Geschichte von Apache Hadoop: Online im Internet: URL: <https://www.geeksforgeeks.org/hadoop-history-or-evolution/>, 9. Jänner 2021
- [LR:Article06] Karen Kriva: Hadoop vs. Spark Debunking the Myth: Online im Internet: URL: <https://www.gigaspaces.com/blog/hadoop-vs-spark/>, 3. Februar 2021
- [LR:Article07] Ion Stoica: Apache Spark and Hadoop: Working Together: Online im Internet: URL: <https://databricks.com/blog/2014/01/21/spark-and-hadoop.html>, 3. Februar 2021
- [LR:Article08] Laurenz Wuttke: Einführung in Apache Spark: Komponenten, Vorteile und Anwendungsbereiche: Online im Internet: URL: <https://datasolut.com/was-ist-spark/>, 4. Februar 2021
- [LR:Article09] Data-Flair: Apache Spark RDD vs DataFrame vs DataSet: Online im Internet: URL: <https://data-flair.training/blogs/apache-spark-rdd-vs-dataframe-vs-dataset/>, 9. Februar 2021
- [LR:Article10] Snehal Nair: PySpark Collaborative Filtering with ALS: Online im Internet: URL: <https://towardsdatascience.com/build-recommendation-system-with-pyspark-using-alternating-least-squares-als-matrix-factorisation-ebe1ad2e7679>, 12. Februar 2021
- [LR:Article11] Gengliang Wang, Reynold Xin und Jules Damji: Benchmarking Apache Spark on a Single Node Machine: Online im Internet: URL: <https://databricks.com/blog/2018/05/03/benchmarking-apache-spark-on-a-single-node-machine.html>, 7. März 2021

[LR:Article12] Laurenz Wuttke: ETL mit Apache Spark: Online im Internet: URL: <https://datasolut.com/etl-mit-spark/>, 13. März 2021

[LR:Article13] Divya Sistla: Top 5 Apache Spark Use Cases: Online im Internet: URL: <https://www.dezyre.com/article/top-5-apache-spark-use-cases/271>, 14. März 2021

[LR:Article14] Jim Scott: A tale of two clusters: Mesos and YARN: Online im Internet: URL: <https://www.oreilly.com/content/a-tale-of-two-clusters-mesos-and-yarn/>, 14. März 2021

[LR:Article15] Sachin P Bappalige: An introduction to Apache Hadoop for big data: Online im Internet: URL: <https://opensource.com/life/14/8/intro-apache-hadoop-big-data>, 21. März 2021

[LR:Article16] Usman Azhar: How to read and write from Database in Spark using pyspark: Online im Internet: URL: <https://medium.com/@usmanazhar4/how-to-read-and-write-from-database-in-spark-using-pyspark-150d39cd8bb72>, 21. März 2021

[LR:GitHub01] GitHub Verzeichnis zum Installieren von Hadoop winutils: Online im Internet: URL: <https://github.com/cdarlint/winutils>, 30. Dezember 2020

[LR:GitHub02] GitHub: pretrained pipelines: Online im Internet: URL: <https://gist.github.com/vkocaman/34e07482de3b660fe251a45278ed7946>, 8. März 2021

[LR:Docs01] Dhruba Borthakur: Docs zu HDFS: Online im Internet: URL: https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html, 2. Jänner 2021

[LR:Docs02] Dokumentation zu YARN: Online im Internet: URL: <https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YARN.html>, 3. Jänner 2021

[LR:Docs03] Dokumentation von Apache HBase: Online im Internet: URL: <https://hbase.apache.org/>, 6. Jänner 2021

[LR:Docs04] Liste mit allen Erweiterungen die direkt von Apache sind: Online im Internet: URL: <https://hadoop.apache.org/>, 6. Jänner 2021

[LR:Docs05] Docs zu Tez: Online im Internet: URL: <https://hadoop.apache.org/>, 9. Jänner 2021

[LR:Docs06] Dokumentation von Apache Spark: Online im Internet: URL: <https://spark.apache.org/>, 5. Februar 2021

[LR:Docs07] Offizielle Apache Spark SQL Dokumentation: Online im Internet: URL: <https://spark.apache.org/sql/>, 7. Februar 2021

- [LR:Docs08] Offizielle Apache Spark Streaming Dokumentation: Online im Internet: URL: <https://spark.apache.org/streaming/>, 10. März 2021
- [LR:Docs09] Offizielle Apache MLlib Dokumentation: Online im Internet: URL: <https://spark.apache.org/mllib/>, 11. März 2021
- [LR:Docs10] Dokumentation von Featurization unter Apache Spark MLlib: Online im Internet: URL: <https://spark.apache.org/docs/latest/ml-features.html>, 12. März 2021
- [LR:Docs11] Dokumentation zu Apache Spark MLlib Pipelines: Online im Internet: URL: <https://spark.apache.org/docs/latest/ml-pipeline.html>, 12. März 2021
- [LR:Docs12] Dokumentation: Apache Spark MLlib Clustering: Online im Internet: URL: <https://spark.apache.org/docs/3.1.1/ml-clustering.html>, 13. März 2021
- [LR:Docs13] Dokumentation: Koalas: Online im Internet: URL: <https://koalas.readthedocs.io/en/latest/>, 13. März 2021
- [LR:Docs14] Dokumentation: Kubernetes: Online im Internet: URL: <https://kubernetes.io/docs/home/>, 14. März 2021
- [LR:Docs15] Dokumentation: GraphX: Online im Internet: URL: <https://spark.apache.org/docs/latest/graphx-programming-guide.html>, 14. März 2021
- [LR:Docs16] Dokumentation: 10 minutes to Koalas: Online im Internet: URL: https://koalas.readthedocs.io/en/latest/getting_started/10min.html, 21. März 2021
- [LR:Video01] Databricks: Project Zen: Making Spark Pythonic — Reynold Xin — Keynote Data + AI Summit EU 2020: Online im Internet: URL: <https://www.youtube.com/watch?v=-vJLTEOdLvA>, 12. März 2021
- [LR:Paper01] Gene M. Amdahl: Validity of the single processor approach to achieving large scale computing capabilities: Online im Internet: URL: <https://www-inst.eecs.berkeley.edu/~n252/paper/Amdahl.pdf>, 30. Dezember 2020
- [LR:Paper02] Research Papers zu Apache Spark: Online im Internet: URL: <https://spark.apache.org/research.html>, 4. Februar 2021
- [SM:Web01] Schubert, Lenhart: Computational Linguistics. Online im Internet: URL: <https://plato.stanford.edu/archives/spr2020/entries/computational-linguistics/>, 2. November 2020
- [SM:Web02] Becker, Lorenz: Morphologie – Die Bausteine von Wörtern (Linguistik). Online im Internet: URL: <https://www.osa.fu-berlin.de/germanistik/beispielaufgaben/morphologie/index.html>, 3. Jänner 2021

[SM:Web03] spaCy Dokumentation: Syntactic Dependency Parsing. Online im Internet: URL: <https://spacy.io/api/annotation#dependency-parsing>, 4. Jänner 2021

[SM:Web04] Reichel, Uwe: Sprachsynthese: Part-of-Speech-Tagging. Online im Internet: URL: https://www.phonetik.uni-muenchen.de/studium/skripten/P6_Sprachsynthese/3_synthese_pos.pdf, Ludwig-Maximilians-Universität München. 4. Jänner 2021

[SM:Web05] Expert System Team: What is Natural Language Processing? Online im Internet: URL: <https://www.expert.ai/blog/natural-language-processing/>, 6. Jänner 2021

[SM:Web06] Sommerlad, Joe: Google Translate: How does the search giant's multilingual interpreter actually work? Online im Internet: URL: <https://www.independent.co.uk/life-style/gadgets-and-tech/news/google-translate-how-work-foreign-languages-interpreter-app-search-engine-a8406131.html>, 17. Jänner 2021

[SM:Web07] McGuire, Nick: How Accurate Is Google Translate in 2018? Online im Internet: URL: <https://www.argotrans.com/blog/accurate-google-translate-2018/>, 17. Jänner 2021

[SM:Web08] Merkert, Pina: Maschinelle Übersetzer: DeepL macht Google Translate Konkurrenz. Online im Internet: URL: <https://www.heise.de/newsticker/meldung/Maschinelle-Uebersetzer-DeepL-macht-Google-Translate-Konkurrenz-3813882.html>, 17. Jänner 2021

[SM:Web09] deepl.com: Der DeepL Übersetzer im Vergleich zur Konkurrenz. Online im Internet: URL: <https://www.deepl.com/quality.html>, 17. Jänner 2021

[SM:Web10] Breindl, Eva: Konnektoren und Funktor-Argument-Struktur. Online im Internet: URL: <https://grammis.ids-mannheim.de/systematische-grammatik/2728>, 18. Jänner 2021

[SM:Web11] monkeylearn.com: Text Classification. Online im Internet: URL: <https://monkeylearn.com/text-classification/>, 18. Jänner 2021

[SM:Web12] Wolff, Rachel: What Is Natural Language Understanding (NLU) & How Does It Work? Online im Internet: URL: <https://monkeylearn.com/blog/natural-language-understanding/>, 18. Jänner 2021

[SM:Web13] Roldós, Inés: Named Entity Recognition: Concept, Guide and Tools. Online im Internet: URL: <https://monkeylearn.com/blog/named-entity-recognition/>, 19. Jänner 2021

- [SM:Web14] Joshi, Prateek: Build a Natural Language Generation (NLG) System using PyTorch. Online im Internet: URL: <https://www.analyticsvidhya.com/blog/2020/08/build-a-natural-language-generation-nlg-system-using-pytorch/>, 19. Jänner 2021
- [SM:Web15] Shrivarsheni: How to Train spaCy to Autodetect New Entities (NER). Online im Internet: URL: <https://www.machinelearningplus.com/nlp/training-customner-model-in-spacy/>, 19. Jänner 2021
- [SM:Web16] Friederici, Angela: Sprache macht den Menschen. Online im Internet: URL: <https://www.mpg.de/9966424/sprachentwicklung-kinder-ueberblick>, 19. Jänner 2021
- [SM:Web17] master-and-more.at: Masterstudium Computerlinguistik - Infos zum Master. Online im Internet: URL: <https://www.master-and-more.at/masterstudium-computerlinguistik.html>, 19. Jänner 2021
- [SM:Web18] Mayo, Matthew: The Main Approaches to Natural Language Processing Tasks. Online im Internet: URL: <https://www.kdnuggets.com/2018/10/main-approaches-natural-language-processing-tasks.html>, 27. Jänner 2021
- [SM:Web19] Couto, Javier: The Definitive Guide to Natural Language Processing (NLP). Online im Internet: URL: <https://monkeylearn.com/blog/definitive-guide-natural-language-processing/>, 27. Jänner 2021
- [SM:Web20] Kang, Eugine: Hidden Markov Model. Online im Internet: URL: <https://medium.com/@kangeugine/hidden-markov-model-7681c22f5b9>, 28. Jänner 2021
- [SM:Web21] Siegmund, David O.: Markovian processes. Online im Internet: URL: <https://www.britannica.com/science/probability-theory/Markovian-processes>, 28. Jänner 2021
- [SM:Web22] exploredatabase.com: Hidden markov model for NLP applications. Online im Internet: URL: <https://www.exploredatabase.com/2020/03/hidden-markov-model-for-nlp-applications.html>, 30. Jänner 2021
- [SM:Web23] ibm.com: Machine Learning. Online im Internet: URL: <https://www.ibm.com/cloud/learn/machine-learning>, 1. Februar 2021
- [SM:Web24] Shewan, Dan: 10 Companies Using Machine Learning in Cool Ways. Online im Internet: URL: <https://www.wordstream.com/blog/ws/2017/07/28/machine-learning-applications>, 1. Februar 2021
- [SM:Web25] Brownlee, Jason: Supervised and Unsupervised Machine Learning Algorithms. Online im Internet: URL: <https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/>, 1. Februar 2021

- [SM:Web26] monkeylearn.com: An Introduction to Machine Learning. Online im Internet: URL: <https://monkeylearn.com/machine-learning/>, 1. Februar 2021
- [SM:Web27] Barba, Paul: Machine Learning (ML) for Natural Language Processing (NLP). Online im Internet: URL: <https://www.lexalytics.com/lexablog/machine-learning-natural-language-processing>, 2. Februar 2021
- [SM:Web28] Cook, Kimberly: Understand The Machine Learning From Scratch For Beginners. Online im Internet: URL: <https://www.houseofbots.com/news-detail/3581-4-understand-the-machine-learning-from-scratch-for-beginners>, 2. Februar 2021
- [SM:Web29] Brownlee, Jason: Logistic Regression for Machine Learning. Online im Internet: URL: <https://machinelearningmastery.com/logistic-regression-for-machine-learning/>, 2. Februar 2021
- [SM:Web30] Chris: Logistische Regression als Machine Learning Algorithmus in Python. Online im Internet: URL: <https://statisquo.de/2018/04/04/logistische-regression-als-machine-learning-algorithmus-in-python/>, 2. Februar 2021
- [SM:Web31] javatpoint.com: Logistic Regression in Machine Learning. Online im Internet: URL: <https://www.javatpoint.com/logistic-regression-in-machine-learning>, 2. Februar 2021
- [SM:Web32] Gandhi, Rohith: Introduction to Machine Learning Algorithms: Linear Regression. Online im Internet: URL: <https://towardsdatascience.com/introduction-to-machine-learning-algorithms-linear-regression-14c4e325882a>, 2. Februar 2021
- [SM:Web33] Ameisen, Emmanuel: How to solve 90% of NLP problems: a step-by-step guide. Online im Internet: URL: <https://blog.insightdatascience.com/how-to-solve-90-of-nlp-problems-a-step-by-step-guide-fda605278e4e>, 3. Februar 2021
- [SM:Web34] Khandelwal, Renu: Word Embeddings for NLP. Online im Internet: URL: <https://towardsdatascience.com/word-embeddings-for-nlp-5b72991e01d4>, 3. Februar 2021
- [SM:Web35] Stecanella, Bruno: What is TF-IDF? Online im Internet: URL: <https://monkeylearn.com/blog/what-is-tf-idf/>, 3. Februar 2021
- [SM:Web36] Nguyen, Eric: Inverse Document Frequency. Online im Internet: URL: <https://www.sciencedirect.com/topics/computer-science/inverse-document-frequency>, 3. Februar 2021
- [SM:Web37] Luber, Stefan: Was ist ein Neuronales Netz? Online im Internet: URL: <https://www.bigdata-insider.de/was-ist-ein-neuronales-netz-a-686185/>, 4. Februar 2021

[SM:Web38] Wuttke, Laurenz: Künstliche Neuronale Netzwerke: Definition, Einführung, Arten und Funktion. Online im Internet: URL: <https://datasolut.com/neuronale-netzwerke-einfuehrung/>, 4. Februar 2021

[SM:Web39] Roell, Jason: Understanding Recurrent Neural Networks: The Preferred Neural Network for Time-Series Data. Online im Internet: URL: <https://towardsdatascience.com/understanding-recurrent-neural-networks-the-preferred-neural-network-for-time-series-data-7d856c21b759>, 5. Februar 2021

[SM:Web40] Sharma, V.: Deep Learning – Introduction to Recurrent Neural Networks. Online im Internet: URL: <https://vinodsblog.com/2019/01/07/deep-learning-introduction-to-recurrent-neural-networks/>, 5. Februar 2021

[SM:Web41] Shreyak: Building a Convolutional Neural Network (CNN) Model for Image classification. Online im Internet: URL: <https://becominghuman.ai/building-a-convolutional-neural-network-cnn-model-for-image-classification-116f77a7a236>, 5. Februar 2021

[SM:Web42] Saha, Sumit: A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way. Online im Internet: URL: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>, 5. Februar 2021

[SM:Web43] Nicholson, Chris: A Beginner's Guide to Word2Vec and Neural Word Embeddings. Online im Internet: URL: <https://wiki.pathmind.com/word2vec>, 26. Februar 2021

[SM:Web44] developers.google.com: Embeddings: Translating to a Lower-Dimensional Space. Online im Internet: URL: <https://developers.google.com/machine-learning/crash-course/embeddings/translating-to-a-lower-dimensional-space>, 26. Februar 2021

[SM:Web45] Semmelmann, Kilian: Was ist Clustering? Definition, Methoden und Beispiele. Online im Internet: URL: <https://datadrivencompany.de/was-ist-clustering-definition-methoden-und-beispiele/>, 13. März 2021

[SM:Web46] Wilentz, Dan: Mmmm Foodporn! A Clustering and Classification Study using Natural Language Processing. Online im Internet: URL: <https://towardsdatascience.com/mmmm-foodporn-a-clustering-and-classification-study-using-natural-language-processing-e2eae8ddefe1>, 13. März 2021

[SM:Web47] Luber, Stefan: Was ist der k-Means-Algorithmus? Online im Internet: URL: <https://www.bigdata-insider.de/was-ist-der-k-means-algorithmus-a-734637/>, 13. März 2021

[SM:Web48] TU München: K-Means. Online im Internet: URL: <https://www-m9.ma.tum.de/material/felix-klein/clustering/Methoden/K-Means.php>, 13. März 2021

[SM:Web49] Wilentz, Dan: Topic Modeling (Vectorize Words and then Reduce Dimensions). Online im Internet: URL: https://github.com/danielwilentz/Cuisine-Classifier/blob/master/topic_modeling/topic_modeling.ipynb, 14. März 2021

[SM:Web50] Ameisen, Emmanuel: Concrete solutions to real problems. Online im Internet: URL: https://github.com/hundredblocks/concrete_NLP_tutorial/blob/master/NLP_notebook.ipynb, 15. März 2021

[SM:Web51] ai-united.de: Recurrent Neural Networks und LSTM. Online im Internet: URL: <http://www.ai-united.de/recurrent-neural-networks-und-lstm/>, 16. März 2021

[SM:Web52] Luber, Stefan: Was ist ein rekurrentes neuronales Netz (RNN)? Online im Internet: URL: <https://www.bigdata-insider.de/was-ist-ein-rekurrentes-neuronales-netz-rnn-a-843274/>, 16. März 2021

[SM:Web53] Luber, Stefan: Was ist ein Long Short-Term Memory? Online im Internet: URL: <https://www.bigdata-insider.de/was-ist-ein-long-short-term-memory-a-774848/>, 17. März 2021

[SM:Web54] Hoory, Ron: Prosody Contour Prediction with Long Short-Term Memory, Bi-Directional, Deep Recurrent Neural Networks. Online im Internet: URL: https://www.researchgate.net/figure/LSTM-gate-with-peephole-connections-showing-the-internal-structure-and-relation-between_fig2_267154161, 17. März 2021

[SM:Web55] Luber, Stefan: Was ist ein Convolutional Neural Network? Online im Internet: URL: <https://www.bigdata-insider.de/was-ist-ein-convolutional-neural-network-a-801246/>, 18. März 2021

[SM:Web56] Becker, Roland: Convolutional Neural Networks – Aufbau, Funktion und Anwendungsgebiete. Online im Internet: URL: <https://jaai.de/convolutional-neural-networks-cnn-aufbau-funktion-und-anwendungsgebiete-1691/>, 18. März 2021

[SM:Web57] inbenta.com: Symbolische KI vs. Machine Learning in der Verarbeitung Natürlicher Sprache. Online im Internet: URL: <https://www.inbenta.com/de/blog/symbolische-ki-vs-machine-learning-in-der-verarbeitung-natuerlicher-sprache/>, 19. März 2021

[SM:Web58] Mishra, Sanatan: Unsupervised Learning and Data Clustering. Online im Internet: URL: <https://towardsdatascience.com/unsupervised-learning-and-data-clustering-eeecb78b422a>, 19. März 2021

- [SM:Web59] spaCy Dokumentation: Language Processing Pipelines. Online im Internet: URL: <https://spacy.io/usage/processing-pipelines>, 24. März 2021
- [SM:Web60] SoMaJo Dokumentation: Introduction. Online im Internet: URL: <https://github.com/tsproisl/SoMaJo>, 24. März 2021
- [SM:Web61] Singh, Falak: Sentiment Analysis Made Easy Using VADER. Online im Internet: URL: <https://analyticsindiamag.com/sentiment-analysis-made-easy-using-vader/>, 25. März 2021
- [SM:Web62] Terry-Jack, Mohammed: NLP: Pre-trained Sentiment Analysis. Online im Internet: URL: <https://medium.com/@b.terryjack/nlp-pre-trained-sentiment-analysis-1eb52a9d742c>, 25. März 2021
- [SM:Web63] Bhavani, Durga: Understanding Named Entity Recognition Pre-Trained Models. Online im Internet: URL: <https://blog.vsoftconsulting.com/blog/understanding-named-entity-recognition-pre-trained-models>, 25. März 2021
- [SM:Article01] Hutchins, John: Retrospect and prospect in computer-based translation. Proceedings of MT Summit VII „MT in the great translation era“, 1999, pp. 30-44
- [SM:Article02] Petrits, Angeliki: The current state of the Commission’s SYSTRAN MT system. European Commission SYSTRAN development team, DG XIII, Luxembourg, pp. 1
- [SM:Article03] Liddy, E.D.: Natural Language Processing. In Encyclopedia of Library and Information Science, 2nd Ed. NY. Marcel Decker, Inc., 2001, pp. 1-15
- [SM:Article04] Prakash M Nadkarni, Lucila Ohno-Machado, Wendy W Chapman: Natural language processing: an introduction. Journal of the American Medical Informatics Association, Volume 18, Issue 5, September 2011, pp. 544-551
- [SM:Article05] Lapp, Edgar: Linguistik der Ironie. Tübinger Beiträge zur Linguistik, Bd. 369, Gunter Narr Verlag Tübingen, ISBN: 3-8233-4224-X, 1997, pp. 24-26
- [SM:Article06] Ganegedara, Thushan: Natural Language Processing with TensorFlow. Packt Publishing Ltd., ISBN: 978-1-78847-831-1, 2018, pp. 1-4
- [SM:Article07] Nitin Indurkhya, Fred J. Damerau: Handbook of Natural Language Processing. Second Edition. Machine Learning & Pattern Recognition Series, CRC Press, Taylor & Francis Group, ISBN: 978-1-4200-8593-8, 2010, pp. 4-5
- [SM:Article08] Hausser, Roland: Grundlagen der Computerlinguistik: Mensch-Maschine-Kommunikation in natürlicher Sprache. Springer-Verlag Berlin Heidelberg, ISBN: 978-3-540-67187-9, 2000, pp. 21

- [SM:Article09] Hackmack, Susanne: Prolog für Linguisten. Universität Bremen, 2000, pp. 6
- [SM:Article10] Clematide, Simon: Programmiertechniken in der Computerlinguistik I. Universität Zürich, Wintersemester 2004/2005, pp. 14
- [SM:Article11] Carl-Engler-Schule: Lineare Regression (Ausgleichsgerade). Carl-Engler-Schule Karlsruhe, 2011, pp. 1
- [SM:Article12] Ruland, Thomas: Einführung in Neuronale Netze. Universität Ulm, 2004, pp. 2-4
- [SM:Article13] LMU München: Clustering. Ludwig-Maximilians-Universität München, 2015, pp. 56-57
- [SM:Video01] edureka!: Natural Language Processing In 10 Minutes | NLP Tutorial For Beginners | NLP Training | Edureka. Online im Internet: URL: <https://www.youtube.com/watch?v=5ctbvkAMQO4>, 6. Jänner 2021
- [AA:Web01] duden.de: Definition von dem Wort Daten. Online im Internet: URL: <https://www.duden.de/rechtschreibung/Daten>, 10. November 2020
- [AA:Web02] techfacts.de: Was sind Daten? Online im Internet: URL: <https://www.techfacts.de/ratgeber/was-sind-daten>, 10. November 2020
- [AA:Web03] Anita Vetter: Wie werden Daten im Internet genutzt? Online im Internet: URL: <https://www.polyas.de/blog/de/allgemein-de/wie-werden-daten-im-internet-genutzt>, 24. November 2020
- [AA:Web04] Michael Radtke: Was ist Big Data? Online im Internet: URL: <https://www.bigdata-insider.de/was-ist-big-data-a-562440/>, 4. Dezember 2020
- [AA:Web05] sas.com: Why Is Big Data Important? Online im Internet: URL: <https://www.sas.com/us/insights/big-data/what-is-big-data.html>, 4. Dezember 2020
- [AA:Web06] Michael Radtke: Was ist Big Data? Online im Internet: URL: <https://www.bigdata-insider.de/was-ist-big-data-a-562440/>, 21. Dezember 2020
- [AA:Web07] Jake Frankenfield: Data Analytics. Online im Internet: URL: <https://www.investopedia.com/terms/d/data-analytics.asp>, 26. Dezember 2020
- [AA:Web08] ComputerWeekly.de: Künstliche Intelligenz (KI). Online im Internet: URL: <https://www.computerweekly.com/de/definition/Kuenstliche-Intelligenz-KI>, 29. Dezember 2020

- [AA:Web09] bosch.com: Die Geschichte der Künstlichen Intelligenz. Online im Internet: URL: <https://www.bosch.com/de/stories/geschichte-der-kuenstlichen-intelligenz/>, 29. Dezember 2020
- [AA:Web10] Dr. Hansjörg Leichsenring: Vier unterschiedliche Arten Künstlicher Intelligenz - Infografik. Online im Internet: URL: <https://www.der-bank-blog.de/typologie-kuenstliche-intelligenz/technologie/29269/>, 30. Dezember 2020
- [AA:Web11] mindsquare.de: Einsatzzwecke von CRUD. Online im Internet: URL: <https://mindsquare.de/knowhow/crud/einsatzzweck>, 31. Dezember 2020
- [AA:Web12] finanzen.net: Daten - Definition. Online im Internet: URL: <https://www.finanzen.net/wirtschaftslexikon/daten>, 1. Jänner 2021
- [AA:Web13] wirtschaftslexikon24.com: Daten. Online im Internet: URL: <http://www.wirtschaftslexikon24.com/d/daten/daten.htm>, 1. Jänner 2021
- [AA:Web14] Flurina Fiona Baumann, Nadine Belinda Brunner, Kim Oliver Tokarski: Digitale Transformation und Unternehmensführung S.223-248. Online im Internet: URL: https://link.springer.com/chapter/10.1007/978-3-658-26960-9_9, 6. Jänner 2021
- [AA:Web15] ig.com: Was ist eine Börse? Online im Internet: URL: <https://www.ig.com/ch/trading-glossar/borse-definition>, 6. Jänner 2021
- [AA:Web16] devstepodia.org: Ablauf der Daten. URL: <https://devopedia.org/artificial-intelligence>, 1. Februar 2021
- [AA:Web17] der-onliner.blogspot.com: Künstliche Intelligenz: 8 Teilbereiche auf einen Blick. Online im Internet: URL: <https://der-onliner.blogspot.com/2019/04/kuenstliche-intelligenz-teilgebiete.html>, 1. Februar 2021
- [AA:Web18] dev.to: CRUD Operations. Online im Internet: URL: <https://dev.to/gitanshuchoudhary/crud-operations-in-modern-javascript-379e>, 1. Februar 2021
- [AA:Web19] thegrammarlab.com: Verarbeitung von Daten. Online im Internet: URL: <http://www.thegrammarlab.com/?nor-portfoliocat=project-guidesTechniken>, 1. Februar 2021
- [AA:Web20] neofonie.de: Datenpyramide. Online im Internet: URL: <https://www.neofonie.de/kuenstliche-intelligenz/crm-text-mining/>, 13. Februar 2021

- [AA:Web21] Dipl.-Ing. (FH) Stefan Luber: Was sind unstrukturierte Daten? Online im Internet: URL: <https://www.bigdata-insider.de/was-sind-unstrukturierte-daten-a-666378/>, 20. Februar 2021
- [AA:Web22] astera.com: Arten von Daten. Online im Internet: URL: <https://www.astera.com/de/Typ/Blog/strukturierte-halbstrukturierte-und-unstrukturierte-Daten/>, 21. Februar 2021
- [AA:Web23] mycourses.aalto.fi: Visuelle Abbildung von dem Zusammenspiel von Big Data, Machine Learning und Systemen. Online im Internet: URL: <https://mycourses.aalto.fi/course/view.php?id=26812>, 25. Februar 2021
- [AA:Web24] harmoniccode.blogspot.com: Beispiel für ein Sankey-Diagramm. Online im Internet: URL: <https://harmoniccode.blogspot.com/2017/12/friday-fun-liii-sankey-plots.html>, 25. Februar 2021
- [AA:Web25] Mustapha Mekhatria: What is a Sankey diagram? Online im Internet: URL: <https://www.highcharts.com/blog/tutorials/what-is-a-sankey-diagram/>, 27. Februar 2021
- [AA:Web26] Thomas Willems: Analytics und KI im Handel: Kaufentscheidungen verstehen. Online im Internet: URL: <https://www.stores-shops.de/technology/analytics-und-ki-im-handel-kaufentscheidung-verstehen/>, 27. Februar 2021
- [AA:Web27] tradistats.com: Informationsgewinnung von Diagrammen und Tabellen erläutern. Online im Internet: URL: <https://tradistats.com/statistik-boersenmonate-im-vergleich/>, 6. März 2021
- [AA:Web28] tradistats.com: Informationsgewinnung von Diagrammen und Tabellen erläutern. Online im Internet: URL: <https://tradistats.com/statistik-boersenmonate-im-vergleich/>, 6. März 2021
- [AA:Web29] tradistats.com: Informationsgewinnung von Diagrammen und Tabellen erläutern. Online im Internet: URL: <https://tradistats.com/statistik-boersenmonate-im-vergleich/>, 6. März 2021
- [AA:Web30] consorsbank.de: Beispiele zur Veränderung des Kursverlaufs in der Börse. Online im Internet: URL: <https://wissen.consorsbank.de/t5/Blog/Wenn-Pressemitteilungen-Kurse-beeinflussen/ba-p/72487>, 9. März 2021
- [AA:Web31] Jesko: GUI - Definition, Bestandteile und Anforderungen. Online im Internet: URL: <https://www.it-talents.de/blog/it-talents/was-ist-gui>, 16. März 2021
- [AA:Web32] Dr. Veikko Krypczyk: Einführung in die Programmierung: Das Aussehen entscheidet. Online im Internet: URL: <https://entwickler.de/online/development/einfuehrung-programmierung-benutzeroberflaechen-238929.html>, 16. März 2021

- [AA:Web33] Dr. Veikko Krypczyk: Einführung in die Programmierung: Das Aussehen entscheidet. Online im Internet: URL: <https://entwickler.de/online/development/einfuehrung-programmierung-benutzeroberflaechen-238929.html>, 16. März 2021
- [AA:Web34] ionos.de: Was ist ein Graphical User Interface (GUI)? Online im Internet: URL: <https://www.ionos.de/digitalguide/websites/web-entwicklung/was-ist-ein-gui/>, 20. März 2021
- [AA:Web35] ionos.de: Welche Anforderungen sollte ein GUI erfüllen? Online im Internet: URL: <https://www.ionos.de/digitalguide/websites/web-entwicklung/was-ist-ein-gui/>, 20. März 2021
- [AA:Web36] ionos.de: Was sind die Bestandteile eines GUI? Online im Internet: URL: <https://www.ionos.de/digitalguide/websites/web-entwicklung/was-ist-ein-gui/>, 20. März 2021
- [AA:Web37] thuatngumarketing.com: Veranschaulichung der Schichten einer GUI. Online im Internet: URL: <https://www.thuatngumarketing.com/graphical-user-interface-viet-tat-gui/>, 20. März 2021
- [AA:Web38] entwickler.de: Abbildung von Designsprachen. Online im Internet: URL: <https://entwickler.de/online/development/einfuehrung-programmierung-benutzeroberflaechen-238929.html>, 20. März 2021
- [AA:Web39] ionos.de: Welche Vor- und Nachteile bietet ein GUI? Online im Internet: URL: <https://www.ionos.de/digitalguide/websites/web-entwicklung/was-ist-ein-gui/>, 22. März 2021
- [AA:Web40] guru99.com: Informationen über das CodeIgniter-Framework. Online im Internet: URL: <https://www.guru99.com/what-is-codeigniter.html>, 23. März 2020
- [AA:Web41] guru99.com: Ablauf des CodeIgniter-Frameworks. Online im Internet: URL: <https://www.guru99.com/what-is-codeigniter.html>, 23. März 2021

Kapitelzuordnung

Kapitelbezeichnung	Seiten	Verantwortliche Person
Zielsetzung	1 - 3	Alle
Allgemeines zu Webcrawling	4 - 22	Nicolas Philipp
Crawler	23 - 50	Nicolas Philipp
Implementation eines Webcrawlers im Projekt AI Börse	51 - 63	Nicolas Philipp
Clustersysteme	64 - 69	Louis Raschbach
Big Data Frameworks, die mit Clustersystem arbeiten	70 - 108	Louis Raschbach
Implementation von Clustersystemen im Projekt AI Börse	109 - 123	Louis Raschbach
Computerlinguistik	124 - 127	Simon Mader
Natural Language Processing	128 - 167	Simon Mader
Analyse von Börsennews im Projekt AI Börse	168 - 184	Simon Mader
Allgemeines über Daten	185 - 204	Alen Asanov
Repräsentation von unformatierten Daten	205 - 244	Alen Asanov

Begleitprotokolle - Asanov

Name	Datum	Thema	Dauer in h
Asanov	03.10.2020	Inhaltsverzeichnis geplant	3,5
Asanov	04.10.2020	Literaturstudium	4
Asanov	06.10.2020	Inhaltsverzeichnis erstellt	2
Asanov	11.10.2020	Literaturstudium	2
Asanov	28.10.2020	Dokument-Setup	3,5
Asanov	10.11.2020	Allgemeines über Daten: Definition	2
Asanov	24.11.2020	Allgemeines über Daten: Wozu braucht man Daten?	1,5
Asanov	25.11.2020	Allgemeines über Daten: Wozu braucht man Daten?	3,5
Asanov	04.12.2020	Allgemeines über Daten: Wozu braucht man Daten?, Big Data	3,5
Asanov	21.12.2020	Allgemeines über Daten: Big Data	3
Asanov	26.12.2020	Allgemeines über Daten: Data Analytics, Korrekturarbeiten	6,25
Asanov	27.12.2020	Allgemeines über Daten: Preptive Selling, Definition	4
Asanov	29.12.2020	Allgemeines über Daten: Definition, Die Geschichte hinter der künstlichen Intelligenz	3,5
Asanov	30.12.2020	Allgemeines über Daten: Künstliche Intelligenz, Arten von KI	4
Asanov	31.12.2020	Allgemeines über Daten: Verarbeitung von Daten, Was ist CRUD?	3
Asanov	01.01.2021	Allgemeines über Daten: Was ist CRUD?, Unterscheidung der Daten, Geschichte der Datenspeicherung	6,5
Asanov	06.01.2021	Allgemeines über Daten: Geschichte der Datenspeicherung, Vergleich Börse	4
Asanov	16.01.2021	Korrekturarbeiten	3,5
Asanov	19.01.2021	Änderung des Inhaltsverzeichnisses	3,25
Asanov	25.01.2021	Allgemeines über Daten: Interpretation und Analyse von Kursverläufen	3,75
Asanov	29.01.2021	Allgemeines über Daten: Interpretation und Analyse von Kursverläufen	2,5
Asanov	30.01.2021	Korrekturarbeiten	4,5
Asanov	01.02.2021	Allgemeines über Daten: Interpretation und Analyse von Kursverläufen	2

Asanov	13.02.2021	Repräsentation von unformatierten Daten: Was sind unformatierte Daten?	3,5
Asanov	21.02.2021	Repräsentation von unformatierten Daten: Struktur der Daten	4,5
Asanov	22.02.2021	Repräsentation von unformatierten Daten: Struktur der Daten	1,5
Asanov	23.02.2021	Korrekturarbeiten	3,75
Asanov	25.02.2021	Repräsentation von unformatierten Daten: Struktur der Daten	4,5
Asanov	26.02.2021	Repräsentation von unformatierten Daten: Struktur der Daten	2,75
Asanov	03.03.2021	Repräsentation von unformatierten Daten: Struktur der Daten, Darstellung von Daten	2
Asanov	06.03.2021	Repräsentation von unformatierten Daten: Darstellung von Daten	4
Asanov	09.03.2021	Repräsentation von unformatierten Daten: Darstellung von Daten	4,5
Asanov	16.03.2021	Repräsentation von unformatierten Daten: Darstellung von Daten, Wie werden Daten dargestellt?	7,5
Asanov	17.03.2021	Repräsentation von unformatierten Daten: Definition einer GUI	2
Asanov	19.03.2021	Repräsentation von unformatierten Daten: Funktion einer GUI, Korrekturarbeiten	6,5
Asanov	20.03.2021	Korrekturarbeiten	3,25
Asanov	21.03.2021	Repräsentation von unformatierten Daten: Funktion einer GUI	6
Asanov	22.03.2021	Repräsentation von unformatierten Daten: Funktion einer GUI	6,5
Asanov	23.03.2021	Repräsentation von unformatierten Daten: Funktion einer GUI, Vergleich an der Projektgruppe AI-Börse	6,5
Asanov	24.03.2021	Repräsentation von unformatierten Daten: Vergleich an der Projektgruppe AI-Börse, Planung	5,75
Asanov	25.03.2021	Korrekturarbeiten, Repräsentation von unformatierten Daten: Was ist Codeigniter	5,25
Asanov	26.03.2021	Repräsentation von unformatierten Daten: Implementation der GUI, Programmierung der GUI	7,25
Asanov	27.03.2021	Korrekturarbeiten, Repräsentation von unformatierten Daten: Programmierung der GUI	6,75
Asanov	28.03.2021	Überarbeitungen der Kapitel	3,5
Asanov	29.03.2021	Danksagung, Zielsetzung, Zusammenfassung	4,5
			181,5

Begleitprotokolle - Mader

Name	Datum	Thema	Dauer in h
Mader	20.10.2020	Inhaltsverzeichnis NLP erstellt	2
Mader	21.10.2020	Inhaltsverzeichnis zusammengeführt	2
Mader	29.10.2020	Dokument-Setup + Vorwort	3
Mader	30.10.2020	Computerlinguistik: Allgemein	4
Mader	02.11.2020	Computerlinguistik: Allgemein	2
Mader	22.12.2020	Computerlinguistik: Geschichte	3,5
Mader	01.01.2021	Computerlinguistik: Geschichte	2
Mader	02.01.2021	Natural Language Processing: Allgemeines + Vorgehensweise	2,5
Mader	03.01.2021	Natural Language Processing: Vorgehensweise	2
Mader	04.01.2021	NLP: Geschichte + Vorgehensweise: Arten von NLP: Lexikalik, Syntaktik, Semantik, Diskurs, Pragmatik	6,5
Mader	05.01.2021	NLP: Geschichte + Lösungsansätze der NLP	3
Mader	06.01.2021	NLP: Allgemeines + Lösungsansätze von NLP + Aufgaben von NLP: Tokenisierung	2,5
Mader	06.01.2021	NLP: Aufgaben von NLP: Tokenisierung + WSD + NER + POS-Tagging + Satz/Synopsis	2,5
Mader	15.01.2021	Klassifikation + Textgenerierung Überarbeitung der Citations	2
Mader	16.01.2021	Aufgaben von NLP: Tokenisierung + POS-Tagging	3
Mader	16.01.2021	Computerlinguistik: Geschichtie; NLP: Aufgaben: Question Answering, Maschinelle Übersetzung	2,5
Mader	17.01.2021	Computerlinguistik: Geschichtie; NLP: Arten -> Ebenen von NLP	2,5
Mader	17.01.2021	Computerlinguistik: Geschichte; NLP: Ebenen: Semantik	1,5
Mader	18.01.2021	NLP: Ebenen: Semantik: Funktor-Argument-Struktur	1,5
Mader	18.01.2021	NLP: Aufgaben: Text-Kategorisierung, Question Answering	3,25
Mader	19.01.2021	NLP: Aufgaben: Textgenerierung, Named Entity Recognition	2,5
Mader	19.01.2021	Finale Ausbesserungen vor der Zwischenabgabe	4

Mader	27.01.2021	NLP: Symbolischer Ansatz	2
Mader	27.01.2021	NLP: Lösungsansätze + Symbolischer Ansatz	2
Mader	28.01.2021	NLP: Statistischer Ansatz: Hidden Markov Model	1
Mader	30.01.2021	NLP: Statistischer Ansatz: Hidden Markov Model	2,75
Mader	31.01.2021	NLP: Statistischer Ansatz	1,25
Mader	01.02.2021	NLP: Statistisches NLP: Machine Learning	4
Mader	02.02.2021	NLP: Machine Learning: Logistische und Lineare Regression	3,5
Mader	03.02.2021	NLP: Aufbereitung der Daten: Bag of Words/One-Hot Encoding, TF-IDF	2,5
Mader	04.02.2021	NLP und KNNs: Was ist ein künstliches neuronales Netz?	2,5
Mader	05.02.2021	NLP: Recurrent & Convolutional Neural Networks	3,5
Mader	26.02.2021	NLP: Word2Vec	1,5
Mader	27.02.2021	Praxisteil: Aufgabenstellung + Erfassung mithilfe eines prozeduralen Ansatzes	2,5
Mader	04.03.2021	Praxisteil: Named Entity Recognition, Sentimentanalyse	3
Mader	10.03.2021	Praxisteil: Named Entity Recognition	2
Mader	11.03.2021	Praxisteil: Überarbeitung	3
Mader	13.03.2021	NLP: Clustering	2
Mader	13.03.2021	NLP: k-Means, Sequence Labeling	3,5
Mader	14.03.2021	NLP: Unsupervised Learning Beispiel	1
Mader	14.03.2021	Generelle Überarbeitung + NLP: k-Means, Unsupervised Learning Beispiel	3,5
Mader	15.03.2021	NLP: Supervised Learning Beispiel	4
Mader	16.03.2021	NLP: Recurrent Neural Network	3
Mader	17.03.2021	Generelle Überarbeitung + NLP: Long Short Term Memory	2
Mader	18.03.2021	NLP: Convolutional Neural Network	3
Mader	19.03.2021	Korrekturen	2
Mader	19.03.2021	Finale Arbeiten vor der Zwischenabgabe	3
Mader	20.03.2021	Finale Korrekturen vor der Zwischenabgabe	4

Mader	22.03.2021	Praxisteil: Named-Entity-Recognition	3
Mader	23.03.2021	Praxisteil: Sentiment-Analyse	3
Mader	24.03.2021	Praxisteil: Prozeduraler Ansatz + generelle Überarbeitung	5
Mader	25.03.2021	Praxisteil: Finale Arbeiten und Korrekturen vor der Zwischenabgabe	5
Mader	26.03.2021	Generelle Überarbeitung	4
Mader	27.03.2021	Überarbeitung Literaturverzeichnis	6
Mader	27.03.2021	begleitprotokolle.tex erstellt	1
Mader	28.03.2021	Begleitprotokolle Formatierung überarbeitet und Kapitelzuordnung eingefügt	3
Mader	28.03.2021	Teile zusammengefügt: Mader + Philipp + Raschbach; Zusammenfassung und Zielsetzung geschrieben	7
Mader	29.03.2021	Überarbeitung der Formatierung	1
Mader	29.03.2021	Überarbeitung Zusammenfassung, Kapitelzuordnung, Fazit	2
Mader	29.03.2021	besprechungsprotokolle.tex erstellt	1
Mader	29.03.2021	Finale Arbeiten vor der finalen Abgabe: Danksagung, Formatierungen, Zusammenfassung	5
			174,25

Begleitprotokolle - Philipp

Name	Datum	Thema	Dauer in h
Philipp	28.09.2020	Brainstorming -> Themenstellung Inhaltsverzeichnis	2
Philipp	01.10.2020	Sammlung von Literatur	1,5
Philipp	05.10.2020	Literaturstudium	2
Philipp	25.10.2020	Dokument-Setup	1,5
Philipp	27.10.2020	Allgemeiner Teil (Indexierung)	3,5
Philipp	28.10.2020	Crawl-Budget, Pagerank	2,5
Philipp	30.10.2020	Hyperlinks	3,5
Philipp	31.10.2020	Formatierung allgemeiner Teil	3
Philipp	01.11.2020	Hyperlinks	2
Philipp	05.11.2020	Hyperlinks	3
Philipp	08.11.2020	Suchmaschinenoptimierung	1,5
Philipp	09.11.2020	Suchmaschinenoptimierung	3,5
Philipp	14.11.2020	Onpage-Optimierung	2
Philipp	15.11.2020	Onpage-Optimierung	2
Philipp	21.11.2020	Onpage-Optimierung	1,5
Philipp	22.11.2020	Offpage-Optimierung	2,5
Philipp	23.11.2020	Offpage-Optimierung	2
Philipp	25.12.2020	Crawler Architektur und Arbeitsweise	3
Philipp	26.12.2020	Crawler Architektur und Arbeitsweise	1,5
Philipp	27.12.2020	Crawler Anforderungen	2
Philipp	29.12.2020	Crawler Arbeitsprozess, URL-Normalisierung	3
Philipp	30.12.2020	URL-Normalisierung	2
Philipp	31.12.2020	URL-Normalisierung	2,5
Philipp	02.01.2021	Crawler Arbeitsprozess, Anforderungen	3

Philip	03.01.2021	Crawler Arbeitsprozess, Anforderungen	4
Philip	04.01.2021	Robots Exclusion Standard/Robots.txt	3
Philip	05.01.2021	Robots.txt Aufbau, Beispiele	3,5
Philip	06.01.2021	Robots.txt Fehler, Spider-Traps	3
Philip	07.01.2021	Sitemap.xml, Überarbeitung Webcrawling	3
Philip	08.01.2021	Überarbeitung Allgemein	1
Philip	09.01.2021	Überarbeitung Allgemein	1,5
Philip	10.01.2021	Überarbeitung, Formatierung	3
Philip	11.01.2021	Überarbeitung, Formatierung	1,5
Philip	12.01.2021	Überarbeitung Allgemein	4
Philip	31.01.2021	Spider-Traps	3,5
Philip	02.02.2021	Webcrawling Frameworks	3
Philip	03.02.2021	Webcrawling Strategien	4,5
Philip	04.02.2021	Anwendungsbereiche von Crawlern	4
Philip	05.02.2021	Webcrawling Frameworks & Tools	3
Philip	10.02.2021	Beispiele für Suchmaschinencrawler: GoogleBot	2
Philip	11.02.2021	Webcrawling Frameworks & Tools	2,5
Philip	13.02.2021	Beispiele für Suchmaschinencrawler: DuckDuckBot	2
Philip	14.02.2021	Beispiele für Suchmaschinencrawler: BingBot	3
Philip	15.02.2021	Beispiele für Suchmaschinencrawler: DuckDuckBot	2,5
Philip	20.02.2021	Webcrawling Frameworks	3
Philip	27.02.2021	Webcrawling Frameworks	1,5
Philip	01.03.2021	Praxisteil Brainstorm & Mockup	2,5
Philip	02.03.2021	Praxisteil Problemstellung	2,5
Philip	05.03.2021	Praxisteil Spezifikation	2,5
Philip	06.03.2021	Praxisteil Webcrawler Arbeitsprozess	3,5

Philipp	09.03.2021	Scrapy Webcrawlingbeispiel	3,5
Philipp	10.03.2021	Goutte Webcrawlingbeispiel	2
Philipp	11.03.2021	Jaunt Webcrawlingbeispiel	3
Philipp	13.03.2021	Finalisierung Theorieteil	3,5
Philipp	14.03.2021	Finalisierung Theorieteil	3,5
Philipp	15.03.2021	Überarbeitung des Theorie Teils	2,5
Philipp	16.03.2021	Grafiken angepasst	3
Philipp	17.03.2021	Überarbeitung generell	2
Philipp	18.03.2021	Praxisteil Implementation Webcrawler	2,5
Philipp	19.03.2021	Praxisteil Implementation Webcrawler	4
Philipp	20.03.2021	Finalisierung Praxisteil Implementation Webcrawler & Automatisierung	6,5
Philipp	23.03.2021	Individuelle Zielsetzung	2,5
Philipp	28.03.2021	Generelle Überarbeitung	4
Philipp	29.03.2021	Danksagung & Allgemeine Überarbeitung	3,5
Philipp	30.03.2021	Endgültige Überarbeitung	3
			178,5

Begleitprotokolle - Raschbach

Name	Datum	Thema	Dauer in h
Raschbach	26.10.2020	Literaturstudium	3,5
Raschbach	30.10.2020	Allgemeiner Teil über Clustersysteme	4
Raschbach	27.12.2020	Arten von Clustersysteme	4,5
Raschbach	28.12.2020	Probleme von Clustersysteme	4
Raschbach	29.12.2020	Überarbeitung des ersten Kapitels	3,5
Raschbach	30.12.2020	Praxisteil Beschreibung des Vorganges	5
Raschbach	31.12.2020	Praxisteil Dokumentation vom Umgesetzten	3
Raschbach	01.01.2021	Hadoop: Geschichte und Allgemeines	2,5
Raschbach	02.01.2021	Hadoop: Funktionsweise	3
Raschbach	03.01.2021	Hadoop: YARN, MapReduce	4,5
Raschbach	04.01.2021	Hadoop: Erweiterungen und ZooKeeper	3
Raschbach	06.01.2021	Hadoop: fertiggestellt	5
Raschbach	08.01.2021	Überarbeitung Hadoop	6
Raschbach	12.01.2021	Kontrollieren und überarbeiten	2,5
Raschbach	13.01.2021	Überarbeiten der Quellen	2,5
Raschbach	01.02.2021	Apache Spark Übersicht	3
Raschbach	03.02.2021	Apache Spark Verhältnis zu Hadoop	4
Raschbach	04.02.2021	Apache Spark Funktionsweise	4,5
Raschbach	05.02.2021	Apache Spark Komponenten GraphX und Spark SQL	3
Raschbach	06.02.2021	Apache Spark Komponenten Streaming	5,5
Raschbach	09.02.2021	Apache Spark Komponenten MLlib Einleitung	3
Raschbach	12.02.2021	Apache Spark MLlib Allgemein	4
Raschbach	06.03.2021	Apache Spark MLlib Funktionsweise	5
Raschbach	07.03.2021	Apache Spark MLlib Teile 1-3 beschrieben	2,5

Raschbach	08.03.2021	Apache Spark MLlib Pipelines Theorie	3
Raschbach	09.03.2021	Apache Spark MLlib Example	5
Raschbach	10.03.2021	Kontrolllesen und verbessern	2,5
Raschbach	11.03.2021	Apache Spark MLlib fertigstellen	3,5
Raschbach	12.03.2021	Apache Spark MLlib Utilities	5
Raschbach	13.03.2021	Apache Spark Fertigstellen	6
Raschbach	14.03.2021	Theorieteil fertigstellen	6,5
Raschbach	15.03.2021	Literaturverzeichnis überarbeiten	2
Raschbach	19.03.2021	Überarbeitung des Theorieteiles	3
Raschbach	20.03.2021	Erklärung von Pipelines im Zuge des Praxisteils	4,5
Raschbach	21.03.2021	Fertigstellung des Praxisteils	8
Raschbach	22.03.2021	Überarbeitung der Diplomarbeit	3,5
Raschbach	23.03.2021	Philipp und Raschbach Teil zusammenfügen	4,5
Raschbach	24.03.2021	Fehlerausbesserung und Zielsetzung	4
Raschbach	25.03.2021	Zielsetzung und Zusammenfassung	3,5
Raschbach	26.03.2021	Korrekturarbeiten	4
Raschbach	27.03.2021	Abgleichen mit den anderen Arbeiten	3,5
Raschbach	28.03.2021	Tätigkeitsbericht und Besprechungsprotokoll ausgearbeitet	5
Raschbach	29.03.2021	Danksagung und Fazit ausgearbeitet und finalisiert	6
			174

Besprechungsprotokolle

Datum	Person(en)	Details
24.06.2020	Alle	Erste Besprechung: Es wurde grob besprochen was das genaue Ziel des Projektes ist und welche Themen daraus kristallisiert werden können.
16.09.2020	Alle	Besprechung der individuellen Zielsetzung und allgemeines Ziel des Projekts
24.09.2020	Alle	Pressemeldung und Diplomarbeitsantrag in der Diplomarbeitsdatenbank besprochen.
07.10.2020	Alle	Einen Termin ausgemacht bis das Inhaltsverzeichnis (+ Seitenanzahl und Terminplan) fertig sein soll. Es wurde auch die insgesamte Seitenanzahl besprochen: mehr als 50 Seiten pro Person
22.10.2020	Alle	Besprechung der Punkte der jeweiligen Personen im Inhaltsverzeichnis.
20.01.2021	Asanov	Feedback zur ersten Zwischenabgabe: Kapitel löschen, da diese nicht zum Thema passt. Inhaltsverzeichnis neu bedenken, auf Themenverfehlung prüfen. Mehr Grafiken einfügen.
20.01.2021	Mader	Feedback zur ersten Zwischenabgabe: schaut alles in Ordnung aus. Besprechung, was noch im Literaturteil vorkommt (Themen, Seitenanzahl) und was im Praxisteil vorkommt.
31.01.2021	Asanov	Schriftliches Feedback zur ersten Zwischenabgabe: Wie bereits besprochen gehört ein Kapitel raus. Arbeit deutlich ansprechender gestalten mittels Grafiken. Bereits besprochene Punkte berücksichtigen.
31.01.2021	Mader	Schriftliches Feedback zur ersten Zwischenabgabe: Literaturarbeiten sind sehr sauber und gut beschrieben. Hardcopies einer realen Anwendung können, wenn möglich, öfters vorkommen. Sie geben dem Leser das Gefühl, dass man nicht nur über theoretische Abhandlungen informiert. Nachfrage, über was im Praxisteil geschrieben wird.

31.01.2021	Raschbach	Schriftliches Feedback zur ersten Zwischenabgabe: die Literaturarbeiten haben alle gepasst.
31.01.2021	Philipp	Schriftliches Feedback zur ersten Zwischenabgabe, Telefonische Klärung der Frage "Was sind typische Dialoge?", Besprechung welche Themen im Theorieteil und im Praxisteil noch folgen werden
18.02.2021	Raschbach	Besprechung, über den Fortschritt der Diplomarbeiten. Es wurde ausgemacht welche Unterpunkte noch besprochen werden sollen und was für Themen dazu gehören.
10.03.2021	Alle	Besprechung der bisher abgegebenen Seiten und der Anzahl der Seiten der nächsten Zwischenabgabe pro Person.
18.03.2021	Alle	Besprechung, auf was bei der finalen Abgabe und beim Binden der Diplomarbeit zu achten ist. Individuelle Zielsetzung und Aufgabenstellung ist wichtig. Im Anhang muss eine Kapitelzuordnung sein. Alle Punkte aus dem Leitfaden müssen erfüllt sein. Formalitäten müssen auf die ganze Diplomarbeit verteilt gleich sein. Physische Abgabe bei der ersten Gelegenheit nach den Osterferien. Besprechung der Zwischenabgaben: In den Überschriften der Praxisteile muss "... im Projekt AI Börse" vorkommen. Es dürfen keine Ich- oder Uns-Bezüge vorkommen.
25.03.2021	Asanov	Feedback zur zweiten Zwischenabgabe: Zitate nochmal anschauen. Die Aussage von Grafiken müssen mittels Beschriftung erkannt werden.
25.03.2021	Mader	Feedback zur zweiten Zwischenabgabe: alles in Ordnung. Abgabe der dritten Zwischenabgabe am Ende des Tages ausgemacht.
25.03.2021	Raschbach	Feedback zur dritten Zwischenabgabe: Hinweis zur umschreiben von einem Ausdruck. Ansonsten positives Feedback bekommen
25.03.2021	Philipp	Feedback zur zweiten und dritten Zwischenabgabe, Besprechung über Vorgangsweise der Abgabe der Quellcodes mittels USB-Stick