

Marcel Gino Faizi
6861531
Bachelor
Informatik
6. Semester
marcel.faizi@stud.uni-frankfurt.de

Bachelorarbeit

Korrelationsanalyse von Kryptowährungskursverläufen basierend auf Tweet Sentiment-Analysis

Marcel Gino Faizi

Abgabedatum: 24.09.2021

Institut für Informatik
Prof. Dr. Alexander Mehler

Zusammenfassung

In dieser Arbeit werden 4,6 Millionen englische Tweets, welche das Keyword „Bitcoin“ enthalten, analysiert und der Zusammenhang zwischen dem Sentiment der Tweets und den Renditen des Bitcoin untersucht. Zur Bestimmung der Sentiment-Klassen werden Text-Klassifizierer mit verschiedenen Ansätzen, darunter auch auf Convolutional Neural Networks und Transformern basierende Modelle, in diesem Kontext evaluiert und optimiert. Es wird außerdem ein Meta-Modell konstruiert, welches beim Problem der Sentiment-Klassifikation von Tweets in drei Klassen {Positiv, Negativ, Neutral} in der betrachteten Domäne besser abschneidet, als die anderen begutachteten Modelle. Bezüglich des Zusammenhangs wird im Speziellen auch der Einfluss von Merkmalen der Tweets und ihrer Verfassern anhand der Distanzkorrelation untersucht.

Inhaltsverzeichnis

| | |
|--|-----------|
| 1. Einleitung | 1 |
| 1.1. Motivation | 1 |
| 1.2. Zielsetzung | 1 |
| 1.3. Aufbau dieser Arbeit | 1 |
| 2. Grundlagen | 2 |
| 2.1. Sentiment Analyse | 2 |
| 2.1.1. Lexikon-basierte Ansätze | 3 |
| 2.1.2. ML-basierte Ansätze | 3 |
| 2.1.2.1. Naiver Bayes Klassifikator | 3 |
| 2.1.2.2. Neuronale Netzwerke | 4 |
| 2.1.2.3. Faltende neuronale Netzwerke | 7 |
| 2.1.2.4. Rekurrente neuronale Netzwerke | 8 |
| 2.1.2.5. Transformer | 9 |
| 2.2. Kryptowährungen | 11 |
| 3. Stand der Forschung | 12 |
| 3.1. Twitter Sentiment und Aktienkurse | 12 |
| 3.2. Twitter Sentiment und Kryptowährungen | 12 |
| 3.3. Abgrenzung | 13 |
| 4. Konzeption | 14 |
| 4.1. Überblick | 14 |
| 4.2. Vorgehen Evaluation | 14 |
| 4.3. Versuchsaufbau | 19 |
| 5. Technische Umsetzung | 25 |
| 5.1. Übersicht | 25 |
| 5.2. Datenspeicherung | 26 |
| 5.3. Datenbeschaffung | 26 |
| 5.3.1. Binance API | 27 |
| 5.3.2. Twitter API | 27 |
| 5.3.3. Twint | 29 |
| 5.4. Vorverarbeitung | 29 |
| 5.5. Annotation | 30 |
| 5.6. Sentiment Analyse | 30 |
| 5.6.1. VADER | 30 |
| 5.6.2. TextBlob | 31 |
| 5.6.3. Stanza | 32 |

| | | |
|-----------|--|-----------|
| 5.6.4. | Huggingface | 32 |
| 5.6.5. | fastText | 33 |
| 5.7. | Performanz | 34 |
| 6. | Darstellung der Ergebnisse | 35 |
| 6.1. | Evaluation der Sentiment-Klassifizierer | 35 |
| 6.2. | Zusammenhang der Zeitreihen | 36 |
| 6.2.1. | Allgemein | 36 |
| 6.2.2. | Filtration mittels Tweet-basierten Merkmalen | 36 |
| 6.2.3. | Filtration mittels User-basierten Merkmalen | 37 |
| 6.2.4. | Kombination der Merkmale | 37 |
| 7. | Diskussion der Ergebnisse | 38 |
| 7.1. | Sentiment-Klassifizierer | 38 |
| 7.1.1. | TextBlob: NaiveBayesAnalyzer | 38 |
| 7.1.2. | TextBlob: PatternAnalyzer | 38 |
| 7.1.3. | Stanza | 38 |
| 7.1.4. | VADER | 38 |
| 7.1.5. | distilBERT | 39 |
| 7.1.6. | BERTweet | 39 |
| 7.1.7. | finBERT | 39 |
| 7.1.8. | fastText Klassifizierer | 39 |
| 7.1.9. | Majority-Vote Klassifizierer | 40 |
| 7.2. | Zusammenhang der Zeitreihen | 40 |
| 7.2.1. | Allgemein | 40 |
| 7.2.2. | Tweet-basierten Merkmalen | 40 |
| 7.2.3. | User-basierten Merkmalen | 40 |
| 7.2.4. | Kombination der Merkmale | 41 |
| 7.3. | Beschränkungen | 41 |
| 8. | Ausblick und Fazit | 42 |
| 8.1. | Mögliche Erweiterungen | 42 |
| 8.2. | Fazit | 43 |
| A. | Ausführliche Ergebnisse | 44 |
| A.1. | Evaluation der Sentiment-Klassifizierer | 44 |
| A.2. | Experiment Teil 1 | 45 |
| A.3. | Experiment Teil 2 | 46 |
| A.3.1. | Tweet-basierte Merkmale | 46 |
| A.3.2. | Autoren-basierte Merkmale | 47 |
| A.4. | Experiment Teil 3 | 48 |
| B. | Programmdokumentation | 50 |
| | Literatur | 53 |

1. Einleitung

1.1. Motivation

Auf Twitter werden verschiedenste Themen diskutiert. Unter diesen Themen sind auch Kryptowährungen, welche mitunter auch sehr emotional geführt werden. Angenommen diese geäußerten Emotionen können finanzielle Entscheidungen von Investoren beeinflussen, dann sollte sich auch ein Zusammenhang zwischen dem Verlauf des Sentiments auf Twitter und den Kursverläufen der Kryptowährungen finden lassen.

1.2. Zielsetzung

Das Ziel dieser Arbeit ist die Untersuchung von aggregierten Twitter-Sentiment-Zeitreihen auf möglichen Zusammenhang mit dem Kursverlauf der Kryptowährung Bitcoin. Dabei werden im Speziellen auch Einflüsse von verschiedenen Merkmalen der Tweets und deren Verfassern auf ebendiesen eruiert. Zu diesem Zwecke werden verschiedene Ansätze zur Sentiment-Klassifikation in der Domäne „Krypto-Twitter“ evaluiert und optimiert. Die sich daraus ergebenden drei Forschungsfragen, welche diese Arbeit versucht zu beantworten, sind:

- Q1: Wie schneiden die verschiedenen Ansätze zur Klassifikation ab bei dem Problem der Sentiment-Klassifikation auf Daten aus der Domäne?
- Q2: Welche Unterschiede bezüglich des Zusammenhangs ergeben sich aus der Wahl des Klassifizierers?
- Q3: Welchen Einfluss haben Eigenschaften von Tweets und ihren Verfassern auf den Zusammenhang?

1.3. Aufbau dieser Arbeit

Kapitel 1 stellt die Motivation und die sich daraus ableitenden Ziele & Forschungsfragen vor. Kapitel 2 führt in die relevanten theoretischen Grundlagen ein. Kapitel 3 zeigt den aktuellen Stand der Forschung auf und grenzt diese Arbeit. Kapitel 4 führt die Vorarbeiten zusammen und beschreibt das gewählte Vorgehen und die getroffenen Annahmen. Kapitel 5 beschreibt die konkrete technische Realisierung und geht auf die verwendeten Sentiment-Klassifikations-Modelle ein. Kapitel 6 führt die erzielten Ergebnisse auf, welche in Kapitel 7 aufgegriffen und diskutiert werden. Kapitel 8 gibt einen Ausblick und resümiert das Erzielte.

2. Grundlagen

2.1. Sentiment Analyse

Twitter Twitter ist ein soziales Netzwerk, welches im Jahr 2006 gegründet wurde und heute weltweit 169 Millionen täglich aktive Nutzer hat. Twitter ermöglicht das Veröffentlichen von kurzen Nachrichten, so genannten "Tweets", welche ursprünglich auf 140 Zeichen limitiert waren, heute allerdings bis zu 280 Zeichen umfassen können. Dabei kann durch die Nutzung des Hashtags „#“ ein Kontext signalisiert werden oder das Zeichen „@“ genutzt werden, um spezifische Nutzer miteinzubeziehen. Tweets können außer Text, Video, Bildern und Audio auch weitere Meta-Informationen beinhalten. Über die Jahre hinweg ist Twitter ein zentraler Platz geworden, an dem Nachrichten, Ankündigungen aus der Privatwirtschaft und von Regierungen, als auch Weltanschauungen und Dinge des täglichen Lebens, geteilt und diskutiert werden. Mit den großen Mengen an diesen kurzen, strukturierten und kontextualisierten Daten ist Twitter einer der Hauptanlaufpunkte für Forschende und Unternehmen geworden, welche versuchen, aus diesen Daten Erkenntnisse zu gewinnen.

Sentiment Nach B. Liu (2015) ist Sentiment-Analyse, auch Opinion-Mining genannt, das Forschungsfeld, welches sich mit der Analyse von Meinungen, Gefühlen, Bewertungen, Einstellungen und Emotionen gegenüber Entitäten und ihren Attributen, welche in Texten geäußert wurden, befasst. Der Begriff der Sentiment-Analyse ist nicht genau definiert und wird deshalb für mehrere Sachverhalte verwendet. Liu schreibt hierzu: „Many related names and slightly different tasks, for example, sentiment analysis, opinion mining, opinion analysis, opinion extraction, sentiment mining, subjectivity analysis, affect analysis, emotion analysis, and review mining, are now all under the umbrella of sentiment analysis.“ Das Sentiment selbst ist nach B. Liu (2015, S.20) definiert, als das zugrundeliegende Gefühl, Einstellung, Bewertung oder Emotion einer Meinung und wird repräsentiert als Tripel (Typ des Sentiments, Orientierung des Sentiments, Intensität des Sentiments). Nach Chaudhuri (2006), zitiert nach B. Liu (2015, S.20f), lässt sich das Sentiment grob in rationales Sentiment, beruhend auf rationalen, greifbaren Überlegungen und emotionales Sentiment einteilen. Das Sentiment kann entweder positiv, neutral oder negativ orientiert sein und die Intensität beschreibt die Stärke der Orientierung des ausgedrückten Sentiments. Sätze, welche Sentiments beinhalten, sind üblicherweise subjektiv. Allerdings können objektive Sätze auch ein Sentiment implizieren. Liu schreibt hierzu (B. Liu 2015, S.2):

For example, based on our commonsense knowledge, we know that "I bought the car yesterday and it broke today" and "after sleeping on the mattress for a month, a valley has formed in the middle" describe two undesirable facts, and we can safely infer that the sentence authors feel negatively about the car and the mattress. Sentiment analysis also studies such objective sentences.

Die Annahmen, die für diese Arbeit diesbezüglich getroffen wurden, werden in Abs. 4.1 beschrieben. Die betrachteten Ansätze zur Bestimmung der Sentiment-Klasse eines Satzes lassen sich grob in Lexikon-, Regel-basierte Ansätze und Ansätze welche auf maschinellem Lernen basieren unterteilen.

2.1.1. Lexikon-basierte Ansätze

Grundlage dieser Ansätze sind Wörter, welche eine klare Wertung implizieren. Beispiele für solche Wörter sind „gut“, „super“, „großartig“, „schlecht“. Eine Sammlung dieser Wörter wird nach B. Liu (2015) als Sentiment-Lexikon bezeichnet und es gibt verschiedene Methoden zur Zusammenstellung solcher Lexika (vgl. dazu auch Abs. 5.6.1). Allerdings sind diese Wörter häufig alleine nicht ausreichend, um zuverlässige Aussagen über das Sentiment eines Satzes treffen zu können. Liu hebt hierzu fünf Fälle hervor, bei denen eine alleinige Beurteilung mittels eines Sentiment-Lexikons unzureichend sein könnte:

1. Ambiguität
2. Sarkasmus
3. Sätze die Wörter aus Sentiment-Lexika enthalten, aber kein Sentiment ausdrücken wollen
4. Sätze die zwar eine Meinung ausdrücken sollen, allerdings kein Wort aus Sentiment-Lexika enthalten
5. Wörter die das Sentiment des Satzes umdrehen

(B. Liu 2015, S.10f, übersetzt MF.)

Der Fall das Wörter das Sentiment eines Satzes umdrehen, kann unter Zuhilfenahme von Heuristiken berücksichtigt werden.(vgl. Hutto und Gilbert 2015)

Andere Fälle wie die Ambiguität von Wörtern hingegen sind schwierig zu lösen und motivieren ihr eigenes Tätigkeitsfeld im Bereich des Natural Language Processing.

2.1.2. ML-basierte Ansätze

In den folgenden Abschnitten werden verschiedene Ansätze des maschinellen Lernens vorgestellt, welche zur Sentiment-Klassifikation von Texten verwendet werden können.

2.1.2.1. Naiver Bayes Klassifikator

Bayes-Theorem Dem naiven Bayes Klassifikator zugrundeliegend ist das Bayes-Theorem. Durch Symmetrie $P(X, Y) = P(Y, X)$ und Produktregel $P(X, Y) = P(Y|X) * P(X)$ der Wahrscheinlichkeitsrechnung ergibt sich das Bayes Theorem als: $P(Y|X) = \frac{P(X|Y)*P(Y)}{P(X)}$,

bzw. mit explizitem Nenner: $P(Y|X) = \frac{P(X|Y)*P(Y)}{\sum_{\text{Observationen}} P(X|Y)*P(Y)}$.(vgl. Bishop 2006, S.15)

Das betrachtete Problem der Sentiment-Klassifikation lässt sich somit wie folgt auffassen: Gegeben dem Eingabe-Vektor $X = \{x_1, x_2, \dots, x_n\}$, wobei x_i das i -te Wort des Satzes der Länge n ist, ist es die Aufgabe diesem eine der $k = 2$ diskreten Klassen C_k zuzuordnen.

Mit dem Bayes Theorem ergibt sich für das Sentiment-Klassifikationsproblem also: $P(C_k|X) = P(C_k) * \frac{P(X|C_k)}{P(X)}$.

Naive Bayes Annahme Unter der Annahme, dass die Observationen, bedingt auf die Klasse, alle unabhängig voneinander sind, lässt sich die bedingte Verteilung über die Klassen bestimmen als:

$$P(C_k|X) \propto P(C_k) * P(x_1|C_k) * P(x_2|C_k) * \dots * P(x_n|C_k)$$

$$= P(C_k) \prod_i^N P(x_i|C_k)$$

(vgl. Bishop 2006, S.202, Fig. 4.81)

Um nun Entscheidungen über die Klassenzugehörigkeit zu treffen, wird die folgende Regel benutzt:

$$\Rightarrow \hat{y}_i = \operatorname{argmax}_{k \in \{1, \dots, k\}} P(C_k) * \prod_i^N p(x_i|C_k)$$

(vgl. Metsis, Androutsopoulos und Paliouras 2006, S.3)

Also ist die zugehörige Klasse für den Satz X die Klasse $k \in \{1, 2\}$, repräsentativ für „positiv“ und „negativ“, für welche die (a posteriori) Verteilung den höchsten Wert annimmt.

2.1.2.2. Neuronale Netzwerke

Die neuronalen Netzwerke (NN) sind Modelle des Deep Learning bzw. noch allgemeiner des Machine Learning. (vgl. Kelleher 2019, S.6, Fig. 1.1) Diese können genutzt werden, um komplexe Muster in Datensätzen zu modellieren.

Sie sind teilweise inspiriert von der Funktionweise biologischer Neuronen im Gehirn. (vgl. Kelleher 2019, S.66f)

NN bestehen aus einem Eingabe-Layer, einem oder mehreren Hidden-Layern und einem Ausgabe-Layer, welche jeweils aus einem oder mehreren Neuronen bestehen. Ein NN, welches mindestens zwei Hidden-Layer hat, kann als tiefes Netzwerk (engl. deep neural network) bezeichnet werden. In Abb. 2.1 sieht man ein einfaches FCNN mit einem Input-Layer, zwei Hidden-Layern und einem Output-Layer.

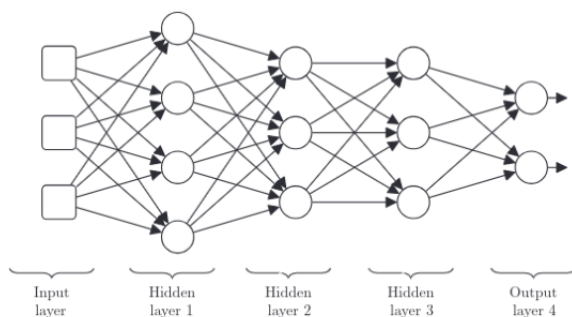


Abbildung 2.1.: (Beispiel Topologie eines neuronalen Netzwerkes, entnommen aus Kelleher (2019).)

Neuronen Neuronen haben eine Menge von Eingaben, welche entweder aus dem Datensatz oder aus einem vorhergehenden Layer stammen. Die Ausgabe z eines Neurons ist die gewichtete Summe aller Eingaben und des Bias-Parameters, auf die ggf. noch eine nicht-lineare Aktivierungsfunktion h angewandt wird. Der Bias-Parameter kann dabei auch in die Gewichtsparameter aufgenommen werden, indem eine zusätzliche Eingabe-Variable x_0 mit einem festen Wert von 1 angelegt wird. Damit entspräche w_0 dem Bias-Parameter. Die Ausgabe lässt sich also formal als $z_k = h(\sum_j w_{kj} * z_j)$ beschreiben. (vgl. Bishop 2006, S.230, Gleichung 5.10) Je nach Position des Neurons ist diese Ausgabe bereits Teil des Endergebnis oder wird an die Neuronen im folgenden Layer weitergegeben.

Aktivierungsfunktion Die erwähnten Aktivierungsfunktionen h ermöglichen es dem NN nicht-lineare Zusammenhänge zu modellieren. (vgl. Kelleher 2019, S.78f) Die jeweilige Aktivierungsfunktion der Neuronen in den Layern kann je nach Format der Eingabe-Daten und der gewünschten Ausgabe des NN variiert werden. (vgl. Kelleher 2019, S.80)

Training Das Lernen des NN lässt sich auffassen als die Suche nach einem Gewichtsvektor w , welcher die Fehlerfunktion $E(w)$ des Netzwerkes für die vorliegenden Trainingsdaten minimiert. Zur Bestimmung von diesem wird ein iteratives numerisches Verfahren angewandt, welches als Gradientenabstiegsverfahren bezeichnet wird. Dabei wird ein Gewichtsvektor mit zufälligen Werten erzeugt und dann mit einem kleinen Wert in negative Richtung des Gradienten der Fehlerfunktion geupdated (siehe Gleichung 2.1).

$$w^{\tau+1} = w^{\tau} - \eta * \nabla E(w^{\tau}) \quad (2.1)$$

(vgl. Bishop 2006, S. 237 und S.240, Gleichung 5.41)

Die gesamte Anzahl von Trainingsschritten bzw. auch Epochen genannt, wird vorher als sogenannter Hyperparameter festgelegt.

Optimierer Die Anpassung des Gewichtsvektors durch das Training kann wie in der Update Gleichung des standardmäßigen Gradientenabstiegsverfahren (Gleichung 2.1), über einen festen Parameter, der sogenannten Lernrate $\eta \in [0, 1]$, erfolgen. Ein großer Wert für diese führt zu einem schnellen Trainingsfortschritt, allerdings können hierbei bessere Lösungen übersprungen werden. Eine zu niedrige Lernrate hat zur Folge, dass es sehr lange dauert bis zur Konvergenz. Optimierer passen diese Lernrate dynamisch an und versuchen so das Risiko des „Steckenbleibens“ in einem lokalen Minimum zu reduzieren, dabei aber immer noch annehmbar schnell zu konvergieren. Die gängigen Optimierer sind Adagrad(Duchi, Hazan und Singer 2011) und Adam(Kingma und Ba 2017). (vgl. Bertschinger 2020)

Regularisierung Ein weiterer wichtiger Faktor bei der Anwendung von neuronalen Netzwerken ist die Fähigkeit des Modells zur Generalisierung. Als Overfitting wird bezeichnet, dass ein Netzwerk für die Trainingsdaten genau lernt gute Aussagen zu treffen, aber auf den Testdaten, welche es zuvor noch nicht gesehen hat, keine zuverlässigen Aussagen treffen kann. Um ein Overfitting des Modells auf die Trainings-Daten zu vermeiden, lassen sich verschiedenen Ansätze der Regularisierung einsetzen:

- **Explizite Regularisierung** (Ng 2004): Beim Minimierungsproblem wird auf die Fehlerfunktion ein Term addiert, welcher für die Anzahl der Gewichte oder die Höhe der Werte der Gewichte steigt.
- **Dropout** (N. Srivastava u. a. 2014): Zufällige Auswahl von Neuronen, welche dann nicht für die Berechnung der Ausgaben des nächsten Layers berücksichtigt werden.
- **Early Stopping** (Goodfellow, Bengio und Courville 2016): Aufhören mit dem Training sobald der Fehler auf dem Validierungsdatensatz für eine vordefinierte Anzahl von Iterationen nicht mehr sinkt.

Wort-Vektoren Um die NN mit Text-Daten verwenden zu können, müssen Wörter jeweils auf Vektoren abgebildet werden. Dabei gibt es die simple Variante des one-hot-Encodings. Dabei werden für die Wörter Vektoren erstellt, deren Länge der Anzahl distinkter Wörter im Datensatz entspricht und an einer Position, welche dem Wort entsprechen soll, eine 1 enthalten und ansonsten eine 0. Häufig werden aber komplexere Varianten, welche selbst auf NN basieren gewählt, um die Vektoren zu erzeugen. Ein Beispiel für diese Varianten sind die CBOW- und Skip-Gram-basierten Wort-Vektoren von Word2Vec nach Mikolov, Chen u. a. (2013). Bei der CBOW(continous bag of words)-Architektur wird ein NN trainiert, mit der Aufgabe ein Wort aus seinem Kontext vorherzusagen (siehe Abb. 2.2, links). Bei der Skip-

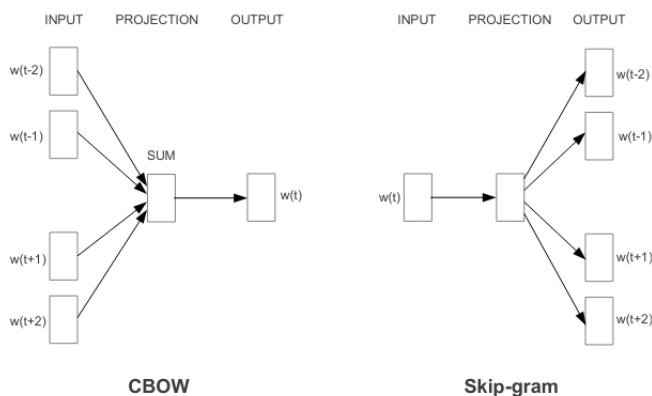


Abbildung 2.2.: Word2Vec: CBOW-Architektur (links), Skip-gram-Architektur (rechts), entnommen aus Mikolov, Chen u. a. (2013).

Gram-Architektur wird die Aufgabe umgedreht, dort wird ein NN trainiert, mit der Aufgabe aus einem Wort, den Kontext vorherzusagen(siehe Abb. 2.2, rechts). Diese Vektoren liegen in einem hochdimensionalen Vektor-Raum, in dem räumliche Nähe semantische Ähnlichkeit der Wörter impliziert. So liegen Wörter mit ähnlichen Anfängen aber unterschiedlichen Endungen nahe beieinander. Des Weiteren sind Vektoroperationen möglich. So ist zum Beispiel $V(\text{König}) - V(\text{Mann}) + V(\text{Frau})$ ungefähr $V(\text{Königin})$. (vgl. Mikolov, Yih und Zweig 2013, S.1) Durch eine Variation der CBOW-Architektur lässt sich auch direkt Textklassifikation und damit auch Sentiment-Klassifikation realisieren. Dabei werden die zu vorhersagenden Wörter

jeweils durch das Label der entsprechend zugehörigen Sentiment-Klasse ersetzt. (vgl. Joulin u. a. 2016, S.2)

2.1.2.3. Faltende neuronale Netzwerke

Eine weitere Klasse, der in dieser Arbeit betrachteten Modelle, sind die faltenden neuronalen Netzwerke bzw. convolutional neural networks (CNN). Diese wurden ursprünglich für die Erkennung von handschriftlichen Ziffern entwickelt (Fukushima 1980, LeCun 1989 nach Kelleher (2019, S.160)). Bei diesen Problemen der Bilderkennung ist das zentrale Ziel, dass das Netzwerk eine Funktion lernt, um das Vorhandensein eines visuellen Merkmals (z.B. eines Auges) zu erkennen.(vgl. Kelleher 2019, S.161) Ausserdem soll es hierbei keine Rolle spielen, wo das Merkmal in dem Bild auftaucht. (LeCun 1989, nach Kelleher (2019, S.161)) Um dieses Ziel zu erreichen erweitern CNN die Architektur normaler NNs um zwei Mechanismen:

Convolutional-Layer Eine der beiden Sonderformen sind die Convolutional-Layer. Die Eingabe des CNN liegt in der Regel als ein-, zwei- oder dreidimensionale Matrix vor. Die Aktivität eines Neurons im Convolutional-Layer wird berechnet, indem ein Filterkernel über die Eingabematrix bewegt wird. Diese ist hierbei definiert als das innere Produkt des Filterkernel, mit dem von ihm anvisierten Ausschnitt der Eingabe-Matrix. Dieser Vorgang wird in Abb. 2.3 gezeigt. Die Eingabe liegt hierbei als 5x5 Matrix vor und der Filterkernel wurde als 3x3 Matrix gewählt. Die Ausgabe 16 ergibt sich beispielsweise als $9 * 0 + 4 * 2 + 1 * 4 + 1 * 1 + 1 * 0 + 1 * 1 + 2 * 0 + 1 * 1 = 16$. Es wird analog zu den Feed-Forward-NN ggf. noch eine nicht-lineare Aktivierungsfunktion auf die Aktivität angewandt. Somit hängt die Aktivität eines Neurons also auch von benachbarten Neuronen ab (z.B. benachbarte Pixel in Bildern oder benachbarte Wörter in Texten).

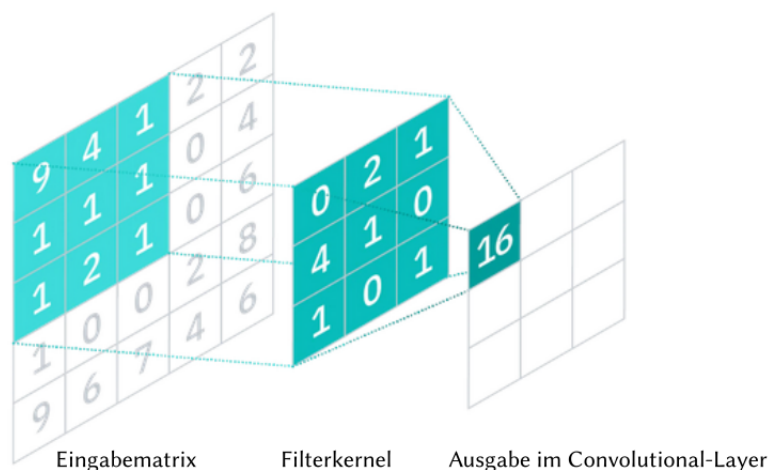


Abbildung 2.3.: Beispiel für Convolutional-Layer, <https://www.ibm.com/cloud/learn/convolutional-neural-networks>, letzter Zugriff: 26.08.2021.

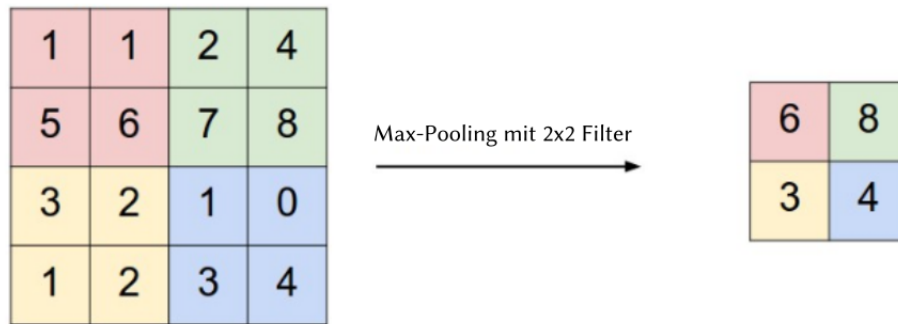


Abbildung 2.4.: Beispiel für Max-Pooling, <https://cs231n.github.io/convolutional-networks/>, zuletzt aufgerufen am 25.08.2021.

Pooling-Layer Im Pooling-Layer werden weniger relevante Informationen verworfen, um die Fähigkeit zur Generalisierung zu verbessern. Die gängigste Variante ist hierbei, das sogenannte Max-Pooling, dabei wird für jeden Filter jeweils nur das Neuron mit der höchsten Aktivität beibehalten und die anderen verworfen. Anders als bei dem Convolutional-Layer wird der Filter beim Pooling allerdings nicht überlappend angewandt, siehe Abb. 2.4.(vgl. Kelleher 2019, S.166) Um die CNN nun für Sentiment-Analyse zu verwenden, werden wie bei den Feed-Forward-NN Wortvektoren für die Token genutzt. Diese Wortvektoren $x_i \in \mathbb{R}^k$ werden konkateniert, so dass sie für einen Satz eine Matrix $\in \mathbb{R}^{n \times k}$ (falls nötig mit 0-Vektoren aufgefüllt) bilden. Darauf folgt ein Convolutional-Layer, ein Max-Pooling-Layer und ein Feed-Forward-NN mit Softmax-Output.(Kim 2014) Ein ähnliches Vorgehen wird in Abb. 2.5 gezeigt.

2.1.2.4. Rekurrente neuronale Netzwerke

Rekurrente neuronale Netzwerke finden in dieser Arbeit keine Anwendung, sollen aber zum Zwecke eines späteren Vergleichs an dieser Stelle kurz eingeführt werden. Diese sind auf die Verarbeitung von Daten in sequentieller Form (Texte, Videos, Zeitreihen) ausgelegt. Das Besondere an dieser Art von Netzwerke ist die Rückkopplung, welche es dem Netzwerk erlaubt, Informationen über mehrere Teilsequenzen t hinweg zu speichern.(vgl. Kelleher 2019, S.170) Es lässt sich auch als Deep-NN vorstellen, welches so tief ist, wie die jeweilige Eingabe-Sequenz lang ist. In der Praxis haben nur die Sonderformen der RNN-Zellen, die LSTM(Long short term memory)-Zellen nach Hochreiter und Schmidhuber (1997) und die GRUs(Gradient recurrent units) nach Cho u. a. (2014) noch Relevanz. Diese Variationen der Architektur versuchen durch Regelung der Informations-Propagation im Netzwerk numerische Instabilität im Trainingsprozess, wie sie bei normalen RNN auftreten kann, zu vermeiden.(vgl. Hochreiter und Schmidhuber 1997, S.1)

Zusammenfassend lässt sich zu den RNN-Modellen und deren Variationen sagen, dass sie anders als Feed-Forward Netzwerke Eingaben variabler Länge zulassen. Auf Eingabe-Sequenzen bis zu einer gewissen Länge können sie Zusammenhänge zwischen den Teil-Sequenzen er-

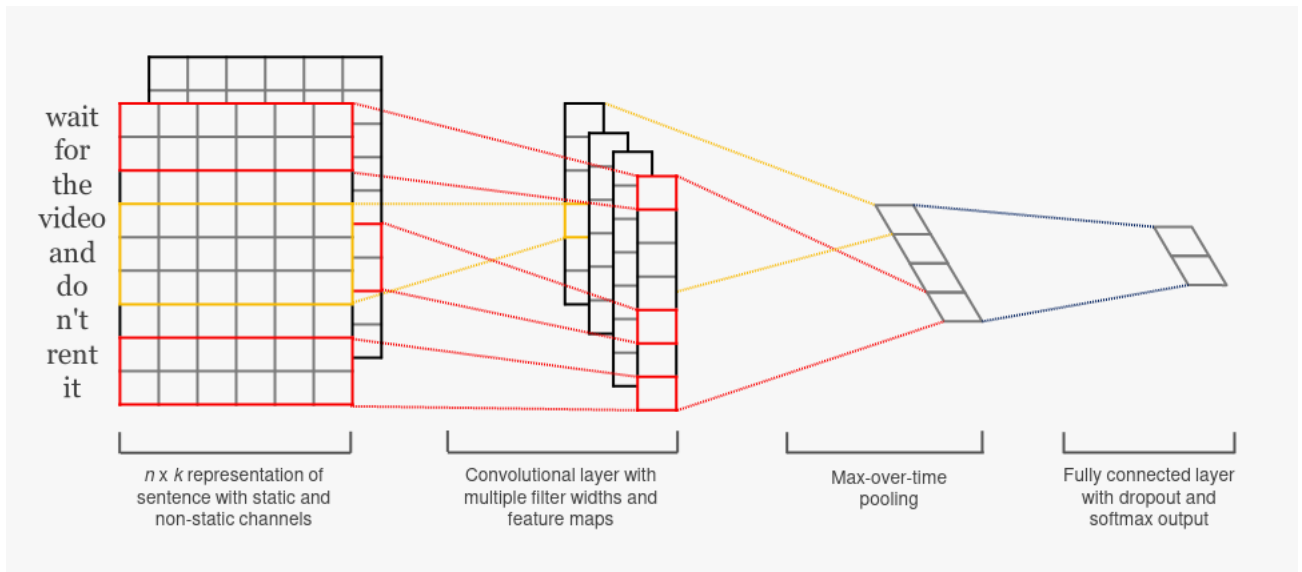


Abbildung 2.5.: Beispiel für CNN-basierte Sentiment-Klassifikation, entnommen aus Kim (2014).

fassen und in ihren Berechnungen berücksichtigen.

2.1.2.5. Transformer

Transformer sind Deep-Learning-Modelle, welche einen Attention-Mechanismus nutzen, um damit die Relevanz einzelner Teile der Eingabedaten unterschiedlich zu gewichten. Die vorher vorgestellten rekurrenz-basierten Modelle haben dahingehend eine Einschränkung, dass sie die Eingaben sequentiell abarbeiten und der Berechnungsschritt für den Zeitpunkt t damit immer von den Berechnungen im vorherigen Zeitpunkt $t-1$ abhängt. Dies hat zur Folge, dass sich die Berechnungen nicht parallelisieren lassen. (vgl. Vaswani u. a. 2017, 1f) Die Transformer Architektur leidet nicht unter dieser Einschränkung, da sie keine Rekurrenz beinhaltet, sondern alleinig auf den eben schon angeführten Attention-Mechanismus setzt, um Abhängigkeiten zwischen den Eingabeteilen zu modellieren. Damit können die Transformer-Modelle auf GPUs (graphics processing units) / TPUs (tensor processing units) große Geschwindigkeitsvorteile erzielen.

BERT In der ursprünglichen Architektur von Vaswani u. a. (2017) konnten Token in den Self-Attention Layern nur von vorigen Token beeinflusst werden.

Für Satz-basierte Aufgaben und besonders für Frage-Antwort-Aufgaben sei dies nach Auffassung der Autoren Devlin u. a. (2019) ungeeignet, da Kontext von beiden Seiten benötigt werde.

Für ihr vorgestelltes Modell BERT nutzen sie zum Training einen Masked-Language-Model-Ansatz. Dabei wird ein Token aus der Sequenz zufällig ausgewählt und versteckt, das Ziel ist es nun, das Token aus dem Kontext vorherzusagen. Des Weiteren wird auch die Vorhersage des nächsten Satzes trainiert. Die Autoren stellen hervor, dass das Training unbedingt auf zu-

sammenhängenden Sätzen stattfinden muss. Aus diesem Grund nutzen sie den Books-Corpus (Zhu u. a. 2015) und englische Wikipedia-Seiten als Trainingsdaten (Zusammen über 3,3 Milliarden Wörter). (vgl. Devlin u. a. 2019, 1f) Um das vortrainierte BERT-Modell für Sentiment-Klassifikation zu nutzen, muss es noch auf die Aufgabe angepasst werden. Dafür wird die Ausgabe des Encoders in ein Feed-Forward Netzwerk gegeben, welches dann letztendlich mittels einer Softmax-Aktivierungsfunktion im Ausgabe-Layer eine Klassifikation erzielt. (Araci 2019, vgl.) Die Transformer-Architektur ist in die gängigen ML-Bibliotheken TensorFlow¹ und PyTorch² integriert. Die in der Arbeit verwendeten Transformer-Modelle stammen aus der Hugging Face-Bibliothek (Wolf u. a. 2020). Dort³ finden sich für verschiedene NLP-Probleme, von der Community, feinabgestimmte Transformer-Modelle.

¹TensorFlow Library, zuletzt aufgerufen am: 01.09.2021

²PyTorch Bibliothek, zuletzt aufgerufen am: 01.09.2021

³Hugging Face Model Hub, zuletzt aufgerufen am: 14.09.2021

2.2. Kryptowährungen

Eine Kryptowährung ist ein digitaler Vermögenswert, welcher durch kryptographische Methoden abgesichert und dezentral gespeichert wird. Der bekannteste Vertreter dieser Kryptowährungen ist Bitcoin, welcher von einem unbekanntem Herausgeber in dem Whitepaper Nakamoto (2009) vorgestellt wurde und als Open-Source Software veröffentlicht wurde. Diese Kryptowährungen werden in der Regel nicht von einer zentralen Instanz, wie einer Zentralbank, herausgegeben, sondern nach dem sogenannten Proof-of-Work-Verfahren, auch schürfen (engl. mining) genannt, erzeugt. Diese Verfahren sind allerdings seit längerer Zeit umstritten, da sie große Mengen von Energie verbrauchen. Im Falle von Bitcoin wird auch noch spezielle Hardware, sogenannte ASICs (Application specific integrated circuits), genutzt, welche nach wenigen Jahren obsolet ist. Als Alternative haben die sogenannten Proof-of-Stake-Verfahren an Popularität gewonnen. Der Handel der Kryptowährungen findet heute größtenteils auf zentralisierten Börsen statt. Allerdings sind diese im Vergleich zu klassischen Wertpapierhandelsplätzen weniger stark reguliert, haben eine deutlich schnellere Abwicklung und ermöglichen einen Handel an 24 Stunden, 7 Tage die Woche. In Tabelle 2.1 sind die fünf größten Kryptowährungen nach Marktkapitalisierung aufgeführt. Die Betrachtungen in dieser Arbeit stützen sich nur auf das Handelspaar der größten Kryptowährung Bitcoin (BTC) und des größten Stablecoins Tether (USDT).

Tabelle 2.1.: Die fünf größten Kryptowährungen nach Marktkapitalisierung (Quelle: coinmarketcap.com , Stand 19.09.2021)

| # | Kryptowährung | Marktkapitalisierung |
|----|--------------------|----------------------|
| 1. | Bitcoin (BTC) | €763,174,429,599 |
| 2. | Ethereum (ETH) | €335,838,980,271 |
| 3. | Cardano (ADA) | €63,305,546,138 |
| 4. | Binance Coin (BNB) | €59,287,436,402 |
| 5. | Tether (USDT) | €58,275,822,030 |

3. Stand der Forschung

In diesem Kapitel werden vorliegende Forschungsergebnisse zum Thema der Sentiment-Analyse in Finanzmarkt-Kontexten aufgeführt.

3.1. Twitter Sentiment und Aktienkurse

Es existiert einige Arbeiten die sich mit diesem Thema beschäftigen.

Rao und S. Srivastava (2012) fanden, dass eine hohe signifikante Korrelation von bis zu 0.88 zwischen Bullishness (einem von ihnen aus dem Sentiment abgeleiteten Wert) der untersuchten Tweets und den Renditen von diversen Tech-Aktien aus dem NASDAQ-100 und DIJA(Dow Jones Industrial Average) besteht.

Sul, Dennis und Yuan (2016) nutzen Twitter-Sentiment um Aktienrenditen vorherzusagen. Die Autoren verwenden hierzu ein Lexikon mit für die Finanzbranche spezifischen Begriffen, um das Sentiment ihrer Tweets zu bestimmen. Dabei finden sie, dass Tweets von Usern (mit geringerer Followeranzahl als der Median von 174) einen signifikanten Einfluss auf die zukünftigen Renditen der von ihnen betrachteten Aktien im S&P500 haben.

Eine Erklärung, die die Autoren für den Zusammenhang zwischen dem Sentiment der Tweets und den Renditen geben, ist, dass das Sentiment der Tweets die Preisveränderungen verursachen. Dies geschehe ihrer Ansicht nach dadurch, dass Investoren die eine Meinung zu einer Aktie haben diese auf Social-Media-Plattformen teilen und damit andere Investoren beeinflussen würden. So hätte sich in (Schoenewolf, 1990; Hatfield et al., 1993, zitiert nach Sul, Dennis und Yuan (2016)) gezeigt, dass Sentiment „ansteckend“ sei und die Kauf-/Verkaufsentscheidungen von Investoren beeinflusst.

3.2. Twitter Sentiment und Kryptowährungen

Trotz der Ähnlichkeiten zwischen dem Handel von Wertpapieren und dem Handel von Kryptowährungen gibt es auch einige gravierende Unterschiede. So stehen hinter Aktien immer Unternehmen an deren wirtschaftlichen Erfolg und Misserfolg man mit dem Halten der Aktie partizipiert. Bei Kryptowährungen hingegen hat die Preisentwicklung keine realwirtschaftliche Untermauerung. Zudem können Kryptowährungen im Gegensatz zu Aktien durchgängig gehandelt werden. Aus diesem Grund werden die Arbeiten zu diesem Thema hier separat betrachtet.

Ante (2021) untersuchte welchen Einfluss die Tweets von Elon Musk auf die Kursverläufe der Kryptowährungen Bitcoin und Dogecoin haben. Dabei fand er für jeden der Tweets darauffolgend ein abnormales Handelsvolumen. Für mehrere betrachtete Zeiträume konnten

signifikante Überrenditen von bis zu 18,99% für Bitcoin und von bis zu 17,31% für Dogecoin festgestellt werden. Er beschreibt allerdings auch, dass nicht alle Tweets abnormale Renditen zur Folge hatten.

Es gibt auch einige Versuche das Sentiment aus Tweets direkt für Preisvorhersagen zu nutzen, wobei die jeweils angeführten Erfolge auch für einen Zusammenhang zwischen dem Sentiment und den Kursen sprechen.

So haben Colianni, Rosales und Signorotti (2015) Tweet-Sentiment für binäre Kursrichtungsvorhersagen des Bitcoins verwendet. Dabei wurden nach Angaben der Autoren mit Tripeln des Sentiments ($pos, neg, neut \in [0, 1]$) als Feature für logistische Regression eine Accuracy von 86% für Tag-zu-Tag-Vorhersagen erzielt.

Li u. a. (2019) nutzen ein Gradient Boosting Tree Model basierend auf Sentiment-Werten von Twitter für die Kursvorhersage der Kryptowährung „ZClassic“.

3.3. Abgrenzung

Diese Arbeit grenzt sich wie folgt von den vorher betrachteten Arbeiten ab.

Einige der Arbeiten nutzen Regel-/Lexikon-basierte Sentiment-Klassifikationsansätze oder beziehen ihre Sentiment-Werte von intransparenten Ansätzen über eine Web-API.

Außerdem fehlen Betrachtungen über die Güte der Sentiment-Klassifikation in der Domäne. Des Weiteren wird in dieser Arbeit kein Modell für Preisvorhersagen konstruiert.

In dieser Arbeit sollen sowohl Regel-/Lexikon-basierte Sentiment-Klassifizierer als auch ML-basierte Ansätze wie die Transformer Modelle betrachtet werden. (Forschungsfrage Q2)

Außerdem sollen diese zuerst auf einem domänenspezifischen Testdatensatz evaluiert werden. (Forschungsfrage Q1)

Die Auswirkungen von Merkmalen der Tweets und deren Verfassern werden anhand der Distanzkorrelation (siehe Abs. 4.3) beurteilt. (Forschungsfrage Q3)

Die Pearsonschen-Korrelationskoeffizienten werden hier der Vollständigkeit halber auch bestimmt, dienen aber nur zur Einordnung in den Forschungskontext und finden ansonsten keine weitere Berücksichtigung.

4. Konzeption

In diesem Kapitel werden die getroffenen Annahmen und das Vorgehen in dieser Arbeit beschrieben.

4.1. Überblick

Es wurden, wie in Abschnitt 2.1 angedeutet, folgende Annahmen bezüglich des Sentiments getroffen:

- **A1:** Jeder Tweet der das Keyword „bitcoin“ enthält, repräsentiert eine Aussage, welche mit Bitcoin in Verbindung steht.
- **A2:** Bei Fakten sollte das Sentiment immer als relativ zum Bitcoin gesehen werden. (vgl. B. Liu 2015, S.42, Beispiel mit Google Aktien)
- **A3:** Konträr zur Annahme von Liu wird ein Tweet nicht als einzelnes Satzklassifikationsproblem gesehen, sondern alle Sätze sollen gleichgewichtet berücksichtigt werden.
- **A4:** URLs sollen nicht berücksichtigt werden.

Die Annahmen dienen einer klaren Definition der Kriterien für die Annotation des Testdatensatzes, sichern die Vergleichbarkeit der Ergebnisse der Evaluation für die Klassifikation und begründen sich wie folgt:

Die Annahme **A1** wird getroffen, da es Tweets geben kann, welche zwar das Schlüsselwort „Bitcoin“ enthalten, aber sich mit einem anderem Thema befassen (z.B. eine Aktie oder eine andere Kryptowährung). Annahme **A2** entspricht der im Grundlagen-Teil genannten Annahme von Liu. Diese ist wichtig, da z.B. auch objektive Aussagen über die Kursentwicklung ein Sentiment implizieren können. Annahme **A3** war nötig, um, wie in Abschnitt 4.2 folgend beschrieben wird, Modelle mit binärer Ausgabe auf drei Klassen abzubilden und um, wie in Abschnitt 4.3 folgend beschrieben, eine „weiche“ Sentiment-Zuweisung zu konstruieren. Annahme **A4** wurde getroffen, um eventuelle Vorteile des BERTweet-Klassifizierers auszugleichen, da dieser wie später erläutert wird, die Fähigkeit hat, URLs bei seinen Analysen zu berücksichtigen. Das Vorgehen in dieser Arbeit lässt sich unterteilen in die Evaluation der Klassifizierer, welche der Beantwortung der Forschungsfrage **Q1** dient und das Experiment mit drei Teilen, das der Beantwortung der Forschungsfragen **Q2** und **Q3** dient.

4.2. Vorgehen Evaluation

Um die Validität der späteren Zusammenhangs-Analysen zu sichern, müssen zuerst die Sentiment-Klassifizierer überprüft werden. Dabei ist es wichtig, ob die Sentiment-Klasse eines Tweets

hinreichend zuverlässig bestimmt werden kann. Die vorliegenden Sentiment-Klassifizierer haben, u.a. wegen der verschiedenen theoretischen Ansätzen, unterschiedliche Ausgabe-Wertebereiche. Auf diese wird im folgenden Kapitel zur technischen Realisierung genauer eingegangen. Da wie Tabelle 4.1 zu entnehmen ist, die meisten der Implementierungen eine Klassifikation in drei Klassen unterstützen, wurde der Ausgabe-Wertebereich für die Evaluation auf die Menge $\{-1,0,1\}$ festgelegt. Dabei repräsentiert -1 ein negatives Sentiment, 0 ein neutrales Sentiment und 1 ein positives Sentiment. Um das gewünschte Ausgabeverhalten

Tabelle 4.1.: Wertebereiche Ausgaben der verwendeten Sentiment-Klassifizierer

| Sentiment-Analyzer | Wertebereich der Ausgabe |
|------------------------------|---|
| TextBlob: PatternAnalyzer | reelwertig $\in [-1,1]$ |
| TextBlob: NaiveBayesAnalyzer | diskret $\in \{\text{pos, neg}\}$ bzw. $p_{\text{pos}}, p_{\text{neg}} \in [0,1]$ mit $p_{\text{pos}}+p_{\text{neg}}=1$ |
| VADER | reelwertig $\in [-1,1]$ |
| Stanza | diskret $\in \{0,1,2\}$ |
| distilBERT | diskret(binär) $\in \{\text{NEGATIVE, POSITIVE}\}$ |
| BERTweet | diskret $\in \{\text{negative, positive, neutral}\}$ |
| finBERT | diskret $\in \{\text{NEG, POS, NEUT}\}$ |
| fastText | diskret(binär) $\in \{\text{NEGATIVE, POSITIVE}\}$ |
| MajorityVote | diskret $\in \{-1,0,1\}$ |

zu erhalten, wurde wie folgt vorgegangen:

VADER Hier wurde die Sentiment-Bewertung `compound_score` verwendet. Die Entscheidungsgrenzen wurden so übernommen, wie es in der zugehörigen Dokumentation von VADER beschrieben wurde. (siehe Algo. 1)

```

Input :  $T$ , ein Tweet;
Output :  $C \in \{-1, 0, 1\}$ , Sentiment-Klasse des Tweets  $T$ .

1  $score = \text{vader.getCompoundScore}(\text{Tweet.text});$ 
2 if  $score < -0.05$  then
3   |  $C = -1$ 
4 end
5 else if  $score > 0.05$  then
6   |  $C = 1$ 
7 end
8 else if  $score \in [-0.05, 0.05]$  then
9   |  $C = 0$ 
10 end
11 return  $C$ ;

```

Algorithmus 1 : Algorithmus zur Klassenzuweisung bei VADER

TextBlob: PatternAnalyzer Das Vorgehen hier war analog zu VADER (Algo. 1). Die Entscheidungsgrenzen wurden hier durch eine qualitative Analyse auf dem Testdatensatz bestimmt. Auf ein Trennen des Tweet-Textes in einzelne Sätze wird hier, wie bei VADER, ebenfalls verzichtet.

TextBlob: NaiveBayesAnalyzer Hier wurde die Wahrscheinlichkeit der Klassenzugehörigkeit als Entscheidungskriterium genutzt. Falls diese für eine der beiden Klassen über 0.7 liegt, wurde diese Klasse gewählt. Ansonsten ist keine hinreichend sichere Aussage möglich und die Klasse Neutral wurde gewählt. Diese Entscheidungsgrenze wurde ebenfalls durch qualitative Analyse auf dem Testdatensatz bestimmt.

Input : T , ein Tweet;

Output : $C \in \{-1, 0, 1\}$, Sentiment-Klasse des Tweets T .

```
1  $p(pos) = nbc(\text{Tweet.text}).p\_pos$ ;  
2  $p(neg) = nbc(\text{Tweet.text}).p\_neg$ ;  
3 if  $p(pos) > 0.7$  then  
4   |  $C = 1$   
5 end  
6 else if  $p(neg) > 0.7$  then  
7   |  $C = -1$   
8 end  
9 else  
10  |  $C = 0$   
11 end  
12 return  $C$ ;
```

Algorithmus 2 : Algorithmus zur Klassenzuweisung bei TextBlob:NaiveBayesAnalyzer

Stanza, BERTweet, finBERT Eine Zuordnung der Sentiment-Werte war hierbei sehr einfach möglich, da die Wertebereiche nur aufeinander abgebildet werden müssen. Das Ausgabeverhalten der Modelle ist, abgesehen von unterschiedlichen Bezeichnern, äquivalent zu dem von Stanza. Die Zuordnung erfolgt bei diesen Modellen, unter Berücksichtigung der jeweiligen Bezeichner, auf die Klasse mit der höchsten Anzahl an Vorkommnissen unter den Satz-Klassifikationen. (siehe Vorgehen für Stanza in Algo. 3, Vorgehen bei BERTweet und finBERT analog.)

distilBERT, fastText Hier wurde ähnlich zu Algo. 3 vorgegangen, die Modellen bieten hier allerdings nur eine binäre Klassifikation. Die Zuordnung erfolgt hier also auf die Klasse mit der höchsten Anzahl an Vorkommnissen unter den Sätzen. Bei Uneindeutigkeit dabei wird die Klasse „Neutral“ zugeordnet. (siehe Vorgehen für distilBERT in Algo. 4, Vorgehen für fastText analog.)

Input : T , ein Tweet;

Output : $C \in \{-1, 0, 1\}$, Sentiment-Klasse des Tweets T .

```
1 Satz_Klassifikationen = stanza.classifications(Tweet.text);
2 Anzahl positiver Klassifikationen pos = 0;
3 Anzahl negativer Klassifikationen neg = 0;
4 Anzahl neutraler Klassifikationen neut = 0;
5 foreach Satz_Klassifikation  $C_i \in$  Satz_Klassifikationen do
6   | if  $C_i == 0$  then
7     |    $neg = neg + 1$ ;
8   | end
9   | else if  $C_i == 1$  then
10    |    $pos = pos + 1$ ;
11   | end
12   | else
13     |    $neut = neut + 1$ ;
14   | end
15 end
16 if  $pos > neg \& pos > neut$  then
17   |    $C = 1$ ;
18 end
19 else if  $neg > pos \& \& neg > neut$  then
20   |    $C = -1$ ;
21 end
22 else
23   |    $C = 0$ ;
24 end
25 return  $C$ ;
```

Algorithmus 3 : Algorithmus zur Klassenzuweisung bei Stanza

Input : T , ein Tweet;
Output : $C \in \{-1, 0, 1\}$, Sentiment-Klasse des Tweets T .

```

1 Satz_Klassifikationen = distilBERT.classifications(Tweet.text);
2 Anzahl positiver Klassifikationen pos = 0;
3 Anzahl negativer Klassifikationen neg = 0;
4 Anzahl neutraler Klassifikationen neut = 0;
5 foreach Satz_Klassifikation  $C_i \in$  Satz_Klassifikationen do
6   | if  $C_i ==$  "NEGATIVE" then
7     |   |  $neg = neg + 1$ ;
8   | end
9   | else if  $C_i ==$  "POSITIVE" then
10  |   |  $pos = pos + 1$ ;
11  | end
12 end
13 if  $pos > neg$  then
14 |   |  $C = 1$ ;
15 end
16 else if  $neg > pos$  then
17 |   |  $C = -1$ ;
18 end
19 else
20 |   |  $C = 0$ ;
21 end
22 return  $C$ ;

```

Algorithmus 4 : Hard-Assignment DistilBERT

MajorityVote Die Sentiment-Klasse entspricht hierbei der Klasse mit den meisten, wie oben beschriebenen konstruierten Zuweisungen. Bei Uneindeutigkeit wird die Klasse „Neutral“ zugewiesen.

Testdatensatz Um die Evaluationen durchführen zu können wurde ein Testdatensatz angelegt, welcher aus 500 zufällig ausgewählten Tweets besteht. Es wurden dabei nur originäre Tweets ohne URLs berücksichtigt. Diese wurden anschließend manuell mit den drei Klassen $\in \{\text{Positiv, Neutral, Negativ}\}$ annotiert.

Evaluationsmaße Anschließend wurden für alle Klassifizierer die folgenden Evaluationsmaße berechnet:

- Precision: $\frac{TP}{TP+FP}$
- Recall: $\frac{TP}{TP+FN}$
- Accuracy: $\frac{TP+TN}{TP+TN+FP+FN}$

(Sebastiani 2002, S. 33f)

- F1-Score: $\frac{2}{\text{recall}^{-1} + \text{precision}^{-1}} = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}$

(Van Rjisbergen 1979, zitiert nach Sebastiani (2002, S.36))

4.3. Versuchsaufbau

Da die Tweets wie eingangs erwähnt abseits der Evaluation nicht zwangsläufig auf diskrete Klassen abgebildet werden müssen und eine Berücksichtigung von Tendenzen zwischen den Klassen sinnvoll erschien, wurde hier abweichend zu dem vorher beschriebenen Vorgehen bei der Evaluation eine weiche Zuweisung des Sentiments verwendet. Das bedeutet die Ausgaben der Klassifizierer müssen nun nicht mehr in der Menge $\{-1, 0, 1\}$ sein, sondern können auch jeden Wert dazwischen annehmen. Für die jeweiligen Klassifizierer wurde dann wie folgt vorgegangen.

VADER Für die weiche Zuweisung von VADER wurde der vorher erwähnte Compound-Wert $\in [-1, 1]$ verwendet.

TextBlob-PatternAnalyzer Um die weiche Zuweisung von dem PatternAnalyzer zu erhalten, wurde wieder äquivalent zu VADER vorgegangen.

TextBlob-NaiveBayesAnalyzer Hier wurde die Differenz $p_{pos} - p_{neg}$ der Wahrscheinlichkeit p_{pos} , dass der Text der positiven Klasse zuzuordnen ist und der Wahrscheinlichkeit p_{neg} für die Zuordnung zur negativen Klasse gewählt. Dieser Wert hat die Eigenschaft, dass er wegen Wahrscheinlichkeitsverteilung ($p_{pos} + p_{neg} = 1$) genau den Wertebereich von $[-1, 1]$ abdeckt. Falls es eine klare negative Zuordnung geben sollte, so würde $p_{pos} = 0$ und $p_{neg} = 1$ gelten, damit wäre die Differenz $p_{pos} - p_{neg} = -1$. Für eine klare Zuordnung zu Positiv analog. Falls Unsicherheit herrscht und die Wahrscheinlichkeit der Klassenzugehörigkeit damit für beide ungefähr gleich ist (rund 0.5), dann würde die Differenz ungefähr den Wert 0 annehmen. Alle Tendenzen in der Sicherheit bezüglich der Zuordnung in die Klassen spiegeln sich also ungefähr proportional zu ihrer Stärke in dieser Differenz wieder.

Stanza, distilBERT, BERTweet, finBERT, fastText Für die anderen Klassifizierer wurde hier, nach dem Vorschlag von D. Baumartz, die weiche Zuweisung als Durchschnitt der den einzelnen Sätzen zugeteilten diskreten Klassen $\{-1, 0, 1\}$ bzw. $\{-1, 1\}$ gewählt. Somit können unterschiedliche Tendenzen zwischen den Sätzen erkannt und berücksichtigt werden.

MajorityVote Die weiche Sentiment-Klassen-Zuweisung wurde hierbei als der Durchschnitt der weichen Zuweisungen aller anderen Klassifizierer gewählt. Das hier der Durchschnitt der weichen Zuweisungen gewählt wurde, anstatt für jeden Satz alle Klassifizierer zu befragen und nach dem Prinzip „winner takes it all“ zu entscheiden und diese Werte zu mitteln, ist durch die Geschwindigkeit begründet (siehe Tab. 5.2). Dies hat, wie sich herausgestellt hat, Implikationen für die Aussagekraft des Wertes für die folgenden Experimente.

Aggregation Um nun einen näherungsweise repräsentativen Sentiment-Wert eines Klassifizierers für eine Zeitperiode zu erhalten, wird jeweils der Durchschnitt der vorher beschriebenen weichen Zuweisungen für alle Tweets aus der jeweiligen Zeitperiode berechnet. Dies wird für alle Zeitperioden in dem insgesamt beobachteten Zeitraum wiederholt und für jeden Klassifizierer eine Zeitreihe der so aggregierten Sentiment-Werte konstruiert. Die Berechnung des aggregierten Werts wird in Gleichung 4.1 formal definiert. Hierbei ist k der jeweilige Klassifizierer, t die betrachtete Zeitperiode und s_k^i die weiche Sentiment-Zuweisung von Tweet i durch k .

$$S_k(t) = \frac{\sum_i^{N_t} s_k^i}{N_t} \quad (4.1)$$

Die betrachteten Perioden-Längen umfassen Stunden und Tage. Auf eine Betrachtung von Minuten wurde verzichtet, da einzelne Minuten mit geringerem Aufkommen von Tweets zu sehr starken Schwankungen führen würden.

Bitcoin-Perioden-Renditen Die Werte der Zeitreihe, welche anschliessend auf Zusammenhang mit den aggregierten Sentiment-Zeitreihen überprüft werden sollen, sind die Perioden-Renditen des Bitcoins. Die Perioden-Rendite wird in Gleichung 4.2 definiert. Dabei ist p_t der Schlusskurs der Periode t , d.h. der letzte handelbare Kurs der Periode. Damit ist p_{t-1} also der

Schlusskurs der Vorperiode.

$$r(t) = \frac{p_t}{p_{t-1}} - 1 \quad (4.2)$$

Zusammenhangsmaße Als Maß des Zusammenhangs wird die Distanzkorrelation verwendet. Die Pearsonsche Korrelation wird zum Vergleich mit vorausgehenden Arbeiten ebenfalls berechnet.

Pearsonsche Korrelation Der Pearsonsche Korrelationskoeffizient ist definiert als: $r_{x,y} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}}$ mit $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$; $\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$. Diese ist ein Maß für den Grad des linearen Zusammenhangs zwischen den Stichproben $X = (x_1, x_2, \dots, x_n)$ und $Y = (y_1, y_2, \dots, y_n)$. (vgl. Hauke und Kossowski 2011, S.1, Notation durch Verfasser angepasst.)

Distanzkorrelation Da die Pearsonschen-Korrelation im Kontext von Finanzmärkten kein hinreichende Maß ist (vgl. Lhabitant 2011), wurde von Prof. A. Mehler die Nutzung der Distanzkorrelation angeregt. Dieses Maß aus Székely, Rizzo und Bakirov (2007) hat den Vorteil, dass es auch nicht-lineare Zusammenhänge erfassen kann. Die Autoren schreiben hierzu: "Distance correlation can also be applied as an index of dependence [...]". (Székely, Rizzo und Bakirov 2007, S.2)

Die empirische Distanzkorrelation für Stichproben der gemeinsamen Verteilung der Zufallsvektoren X, Y ist definiert als:

$$dCor_n(X, Y) = \begin{cases} \frac{dCov_n(X, Y)}{\sqrt{dVar_n(X)dVar_n(Y)}}, & \text{Wenn } dVar_n(X)dVar_n(Y) > 0, \\ 0, & \text{falls } dVar_n(X)dVar_n(Y) = 0. \end{cases}$$

Wobei $dCov_n(X, Y) = \frac{1}{n^2} \sum_{k,l=1}^n A_{kl}B_{kl}$

mit $A_{kl} = |X_k - X_l| - \frac{1}{n} \sum_{l=1}^n |X_k - X_l| - \frac{1}{n} \sum_{k=1}^n |X_k - X_l| + \frac{1}{n^2} \sum_{k,l=1}^n |X_k - X_l|$
 , für $k, l \in \{1, \dots, n\}$; $B_{k,l}$ analog.

(Székely, Rizzo und Bakirov 2007, S.5f, Notation vom Verfasser angepasst.)

Die Distanzkorrelation erfülle folgende Eigenschaften:

- $0 \leq dCor(X, Y) \leq 1$
- $dCor(X, Y) = 0 \Leftrightarrow X$ und Y unabhängig.

(Székely, Rizzo und Bakirov 2007, S.1, übersetzt MF, Notation vom Verfasser angepasst.)

Versuch: Teil I Der erste Teil der Versuchs befasst sich nun mit der Frage, ob sich zwischen den wie oben beschrieben angelegten aggregierten Sentiment-Zeitserien und den Bitcoin-Rendite-Zeitserien mittels den drei Maßen ein Zusammenhang feststellen lässt. Die Ergebnisse für die Distanzkorrelationen dienen als Baseline für den folgenden zweiten Teil des Versuchs.

Versuch: Teil II Der zweite Teil des Versuchs erweitert dies um die Dimension von Merkmalen der Tweets und deren Verfassern. Dabei wird jeweils nur eine Teilmenge des Twitter-Datensatzes verwendet, welche alle ein bestimmtes Merkmal aufweisen. Die Bildung der Teilmenge wird hier aufgrund der Implementierung auch als Filter bezeichnet. Das heißt alle Tweets, die durch den Filter erfasst werden, sind in der jeweiligen Teilmenge. Die Merkmale, die in dieser Arbeit Berücksichtigung finden, sind zu unterscheiden in Tweet-basierte Merkmale und Autoren-basierte Merkmale. Die Merkmale und die erwarteten Auswirkungen sind in den Tabellen 4.2, 4.3 beschrieben.

Versuch: Teil 3 Der dritte Teil des Versuchs erweitert dies nun noch mal um die Kombination dieser Merkmale. Merkmale, welche allerdings nur Ausprägungen des selben Grundmerkmals sind, können nicht miteinander kombiniert werden, so z.B.: User hat mehr als 1.000 Follower und User hat mehr als 100.000 Follower. Hier werden wie in den vorherigen Teilen für die induzierten Teilmengen die aggregierten Sentiment-Werte berechnet und die Zusammenhangsmaße werden berechnet.

Tabelle 4.2.: Tweet-basierte Merkmale & erwartete Auswirkung auf die Distanzkorrelation.

| Merkmal | <i>Erwartete Auswirkung</i> |
|---|---|
| Tweet enthält eine URL | Dieses Merkmal wurde gewählt um den Einfluss von Inhalten zu beurteilen, welche von Twitter wegführen. Eine kurze Einsichtnahme in diese Tweets hat gezeigt, dass es sich hauptsächlich um Links auf Web-Artikel traditioneller Medien und auch auf vermutlich nicht besonders seriöse Investmentmöglichkeiten handelt. |
| Tweet enthält das Wort "free" | Eine Einsichtnahme in die Tweets, die dieses Wort bzw. Variationen „FREE“, „Free“ enthält, hat gezeigt, dass dabei häufig auf Gewinnspiele verwiesen wird. Die Annahme ist, dass die Tweets keinen bzw. geringeren Zusammenhang mit den Preisen aufweisen. |
| Bei dem Tweet handelt es sich um einen Retweet | Ein Retweet kann Zustimmung zu dem im originalen Tweet geäußerten Sentiment zeigen. Allerdings ist hierbei zu beachten, dass der Retweet zu dem originären Tweet nicht in direktem zeitlichen Zusammenhang steht bzw. die Methodik dieser Arbeit dafür keine Korrektur anstrebt. Außerdem könnten diese von anderen Nutzern besonders leicht als Rauschen betrachtet werden und in der Menge untergehen. Aus diesen Gründen wird davon ausgegangen, dass sich das Sentiment geringer auswirkt und der Zusammenhang sinkt. |
| Bei dem Tweet handelt es sich um eine Antwort | Es wird angenommen, dass eine Antwort ein geringeren Einfluss aufweist, da sie vermutlich von anderen Benutzern insgesamt weniger Aufmerksamkeit bekommt. |
| Bei dem Tweet handelt es sich um einen Zitat | Ein Zitat ist vom Prinzip eine Kombination der Antwort und des Retweet. Aus diesem Grund wird ebenfalls ein abnehmender Zusammenhang vermutet. |
| Bei dem Tweet handelt es sich um einen normalen Tweet | Es wird vermutet, dass diese Teilmenge von Tweets den größten Einfluss haben, da sie von den obigen 3 Kategorien von Tweets bereinigt sind, welche annahm gemäß, alle negative Auswirkungen auf die Distanzkorrelation haben. |

Tabelle 4.3.: Autor-basierte Merkmale & erwartete Auswirkung auf die Distanzkorrelation.

| Merkmal | <i>Erwartete Auswirkung</i> |
|--|---|
| Anzahl der Follower des Autors | Die Annahme hierbei ist, dass der Post eines Nutzers der mehr Follower hat potentiell von mehr anderen Usern gesehen wird und damit einen höheres Gewichtung des Sentiments zur Folge hat. |
| Anzahl der Follower des Autors | Es wird angenommen, dass Nutzer, die mehr anderen Nutzern folgen, potenziell auch mehr Personen folgen, welche Informationen zu Kryptowährungen teilen. Damit hat der Nutzer potenziell einen besseren Überblick und deshalb sollten die Sentiments des User mehr Gewicht haben. |
| Zeitdistanz zwischen dem Posten des Tweets und der Erstellung des Accounts | Hier werden verschiedene Grenzen genutzt. Es wird angenommen, dass Tweets die von Autoren gepostet worden sind, welche ihren Account erst vor kurzem erstellt haben, oft Spam posten. Des Weiteren wird angenommen, dass User mit älteren Accounts durch mehr Erfahrung ein besseres Bild des Marktes haben und deren Einschätzungen ein größeren Zusammenhang aufweisen. |
| Anzahl der Likes des Autors | Keine Auswirkung auf den Zusammenhang erwartet. |
| Anzahl der Tweets des Autors | Ebenfalls keine Auswirkung auf den Zusammenhang vermutet. |

5. Technische Umsetzung

In dem folgenden Kapitel werden die verschiedenen Schritte von der Beschaffung der Tweets und Kurse, über die Vorverarbeitung und Sentiment-Klassifikation der einzelnen Tweet-Texte, bis hin zur Analyse der aggregierten Sentiment- und Kurs-Rendite-Zeitreihen in ihrer technischen Umsetzung beleuchtet. Dabei wird näher auf die verwendeten Sentiment-Analyse Werkzeuge eingegangen und unter Rückführung auf Abs. 2.1 die konkret zugrunde liegenden Implementationen (verwendete Korpora, Datensätze, Heuristiken etc.) dieser thematisiert.

5.1. Übersicht

Die entwickelten Python-Skripte und die zugehörigen Jupyter Notebooks umfassen folgende Funktionalitäten:

- **Import:** Importierung vorhandener Tweet-, Annotations- und Kurs-Daten in den jeweils gängigen Dateiformaten.
- **Sentiment-Analyse:** Analysiert Sentiment der Tweets mithilfe der verschiedenen Werkzeuge. Dabei ist die Analyse der Texte aus den Tweet-JSON Dateien auch portabel, d.h. unabhängig von der Datenbank, möglich.
- **Visualisierung:** Visualisierungen der Ergebnisse generieren.
- **Evaluation:** Berechnet Precision, Recall, F1-Score, Accuracy für jeden der Klassifizierer.
- **API-Zugriff:** Datensatz um Autordaten erweitern.
- **Zusammenhangs-Analyse:** Berechnet Distanzkorrelation und p-Korrelationskoeffizienten.

Die Jupyter Notebooks dienen dabei als Alternative zu einem Kommandozeilen-Interface und sollen mit integrierter Anleitung eine bessere Nutzererfahrung bieten. Das sich daraus ableitende Use-Case-Diagramm lässt sich Abb. 5.1 entnehmen.

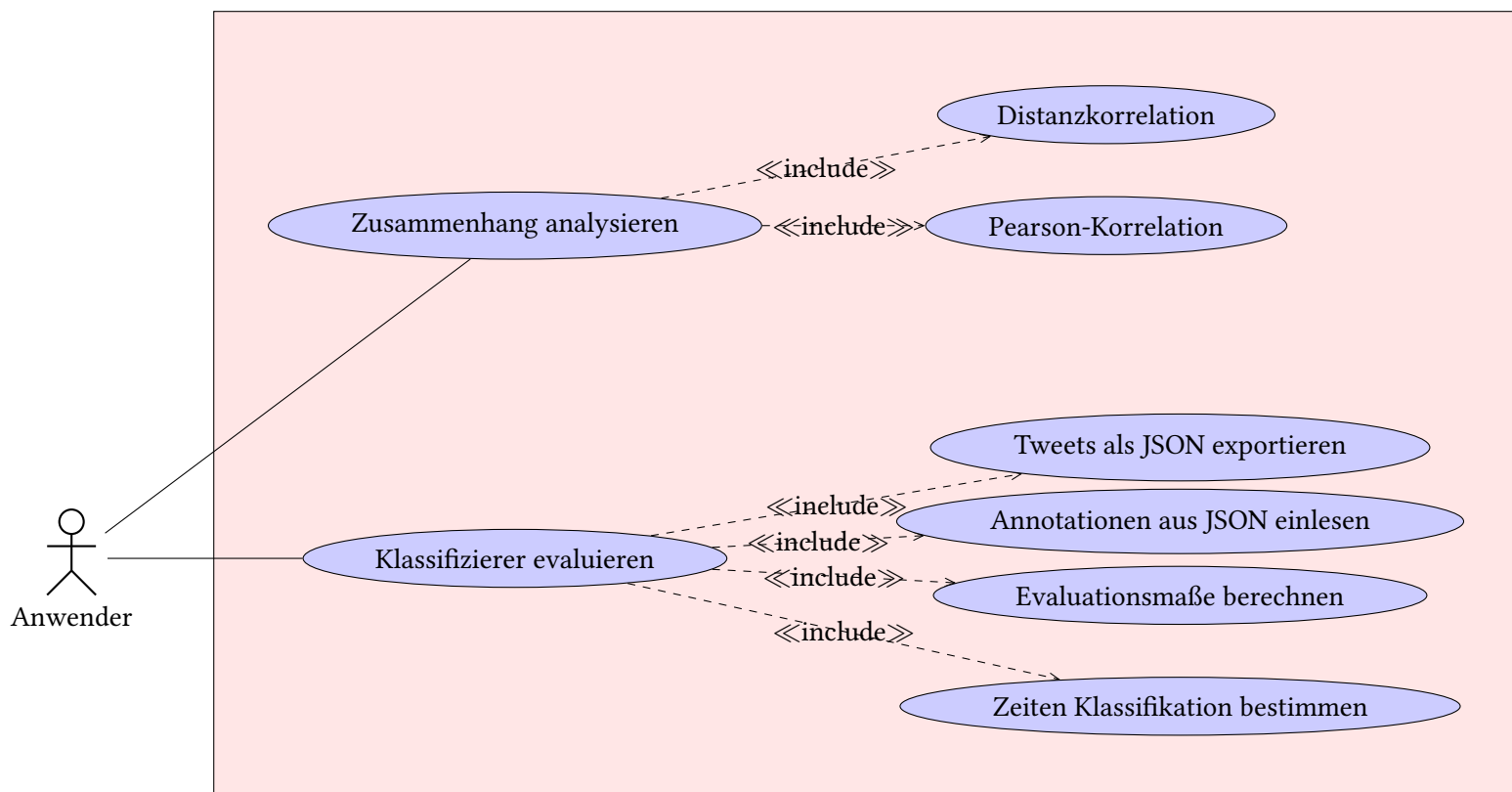


Abbildung 5.1.: Grobes Anwendungsfalldiagramm.

5.2. Datenspeicherung

Zur Speicherung der bei der Analyse anfallenden Daten wird eine MongoDB-Instanz verwendet. Die Datenbank unterteilt sich in 5 Collections, eine für die Tweet-Daten, eine für Autorendaten, eine für Kurs-Daten und zwei für die Ergebnisse der Analysen. Die Ergebnisse der Analysen aus den Experimenten 2 & 3 werden in der Datenbank gespeichert, um unnötige Neuberechnung zu vermeiden, da diese sehr zeitaufwendig sind (ca. 30 Min. und 6 Std.). Die Autorendaten werden nur bei dem Vervollständigen des Datensatzes gebraucht und danach in die jeweilig betroffenen Tweet-Datenpunkte aufgenommen, um häufige zeitaufwendige JOIN-Operationen zu vermeiden.

5.3. Datenbeschaffung

Der Datensatz umfasst den Zeitraum von dem 09. Februar 2021 bis zu dem 30. Februar 2021. Der Auswahl dieses Zeitraums liegt keine direkte Intention zugrunde, sondern dieser ist eine Teilmenge des ursprünglich gewählten Zeitraums vom 01.01.2020 bis zum 01.03.2021, welcher durch das unerwartet hohe Tweet-Aufkommen und Beschränkungen der Twitter-API zustande gekommen ist.

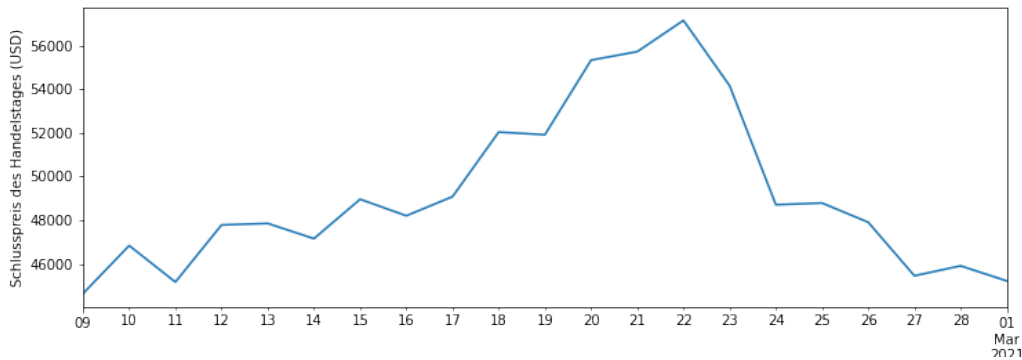


Abbildung 5.2.: Schlusspreise der Handelstage im Zeitraum.

5.3.1. Binance API

Zum Abrufen der Kurse wurde die REST-API, der zur Zeit vom Handelsvolumen größten Kryptowährungsbörse Binance, genutzt. Diese hat u.A. einen Endpunkt für die Daten eines Kerzenkursverlaufdiagramms. Im Gegensatz zu der Twitter-API unterliegt diese keinen expliziten Einschränkungen des Kontingents und der Abfrage-Rate. Des Weiteren ist dieser API-Endpunkt frei zugänglich und kann so von jedem, ohne weitere Authentifizierung, genutzt werden. Der API-Endpunkt liefert einen UNIX-Timestamp¹ und dazu zugehörige: Öffnungskurs der Periode, höchster Kurs der Periode, niedrigster Kurs der Periode, Schlusskurs der Periode, Timestamp vom Ende der Periode, anfallendes Volumen der Periode in Bitcoin, anfallendes Volumen der Periode in USDT, Anzahl der vollzogenen Handel in der Periode und Volumina der vollzogenen Transaktionen in USDT & Bitcoin. Der Datensatz wird in die vorher beschriebenen designierte Collection „prices“ gespeichert. Abbildung 5.2 beschreibt den Verlauf der so erhobenen Preis-Daten in dem beobachteten Zeitraum. In Abb. 5.3 sind die für die spätere Analyse des Zusammenhangs relevanten Veränderungen der Schlusspreise über den Zeitraum von dem 09. Februar 2021 bis zum 30. Februar 2021 hinweg aufgezeigt.

5.3.2. Twitter API

Um den Datensatz mit den Tweets zu erhalten wurde die API von Twitter verwendet. Diese bietet die Möglichkeit einen Query zu stellen, mit einem oder mehreren Schlüsselwörtern, die die Tweets enthalten sollen, einen Zeitraum anzugeben, eine bestimmte Sprache zu selektieren und auch nach geographischer Lage der Tweet-Ersteller zu filtern. An dieser Stelle wird nur die Option nach dem Keyword „Bitcoin“ zu suchen verwendet. Da der Zugriff auf historische Daten akademischen Forschern und zahlenden Kunden vorbehalten ist, wurden der Datensatz freundlicherweise zur Verfügung gestellt. Insgesamt enthält dieser 5.494.508 Tweets. Die Verteilung der Tweets über den Zeitbereich hinweg lässt sich Abb. 5.5 entnehmen. In Abb. 5.4 lässt sich außerdem noch die durchschnittliche Verteilung der

¹Anzahl der Sekunden seit dem 01.01.1970 UTC.

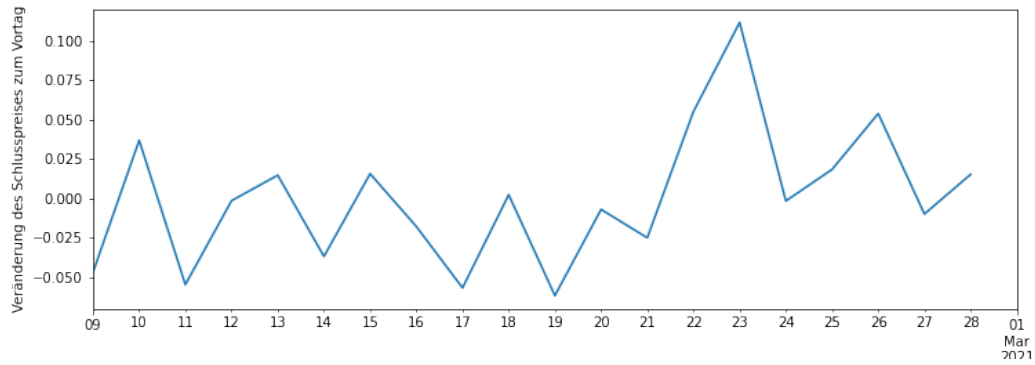


Abbildung 5.3.: Veränderungen der Schlusspreise zum Vortag.

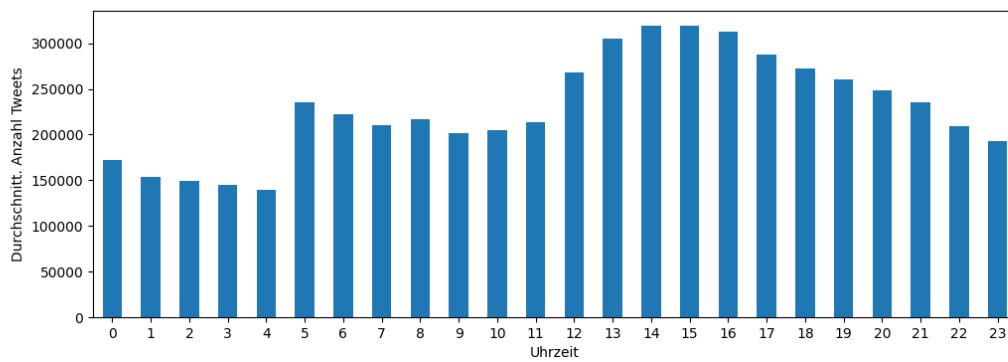


Abbildung 5.4.: Durchschnittliche Verteilung der Anzahl von Tweets über den Tagesverlauf.

Tweets über den Tagesverlauf hinweg entnehmen. Von den so erhaltenen Tweets wurden insgesamt 4.606.209 in englischer Sprache verfasst und wurden somit für die Erstellung der aggregierten Sentiment-Zeitreihen berücksichtigt.

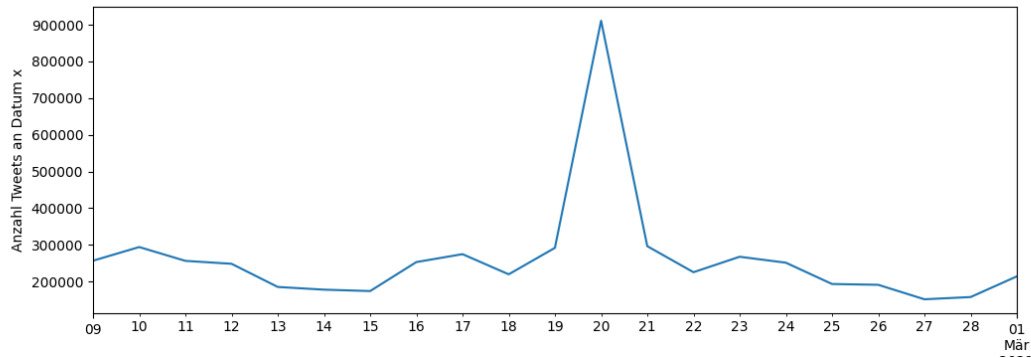


Abbildung 5.5.: Verteilung der Tweets über den Zeitraum.

5.3.3. Twint

Um die Daten über die Autoren der Tweets zu erhalten, wurde die Open-Source Python-Bibliothek Twint genutzt. Diese emuliert mithilfe von BeautifulSoup4 einen Webbrowser und kann so Informationen aus der HTML-Struktur der Twitter-Seite extrahieren. Damit wurden für jeden Nutzer bzw. jede author_id aus dem Tweet-Datensatz die folgenden Informationen abgefragt:

- Anzahl Tweets gesamt
- Anzahl Likes gesamt
- Anzahl Follower
- Anzahl Followees
- Wurde von Twitter verifiziert?
- Datum des Beitritts zu Twitter

5.4. Vorverarbeitung

Zur Vorverarbeitung der Tweet-Texte wurden mehrere Funktionen implementiert.

- **remove_newline**: Entfernt alle Zeilenumbrüche aus dem Text.
- **remove_url**: Entfernt alle URLs aus dem Text.
- **lowercase**: Gibt den Text in Kleinbuchstaben zurück.

5.5. Annotation

Die Annotationen der Tweets wurden mithilfe des Open-Source Tools doccano durchgeführt. Dieses Tool erhält eine JSON-Datei mit einem Array von Tweet-Texten. Zuerst muss über das Webinterface eine Annotationsaufgabe angelegt werden. Dabei wurden der Aufgaben-Typ Textklassifikation gewählt und die Label $\in \{\text{Negativ, Neutral, Positiv}\}$ gewählt. Daraufhin wird eine JSON-Datei mit den Tweet-Texten erstellt und hochgeladen. Diese Tweets wurden dann über das Webinterface annotiert und konnten anschließend mit den jeweiligen Annotationen vervollständigt wieder heruntergeladen werden.

5.6. Sentiment Analyse

Im folgenden Abschnitt werden die verschiedenen Python-Module beschrieben, die zur Analyse des Sentiments der Tweets in dieser Arbeit verwendet wurden. Bei der Auswahl der Werkzeuge wurde darauf geachtet, dass Vertreter beider im Grundlagen-Teil vorgestellten Klassen abgedeckt werden.

Tabelle 5.1.: Ansätze der verschiedenen Sentiment-Analyzer Implementationen.

| Sentiment-Analyzer | <i>Ansatz der Implementation</i> |
|------------------------------|--|
| TextBlob: PatternAnalyzer | Sentiment-Lexikon- und Heuristik-basiert |
| TextBlob: NaiveBayesAnalyzer | Basierend auf Naivem Bayes Klassifikator |
| VADER | Sentiment-Lexikon- und Heuristik-basiert |
| Stanza | CNN-basiert |
| distilBERT | Transformer-basiert |
| roBERTa | Transformer-basiert |
| BERTweet | Transformer-basiert |
| finBERT | Transformer-basiert |
| fastText | Wortvektor-basiert |

5.6.1. VADER

VADER (Valence Aware Dictionary for sEntiment Reasoning) ist im Gegensatz zu den anderen in dieser Arbeit verwendeten Werkzeugen, das Einzige, welches speziell auf Sentiment-Analyse ausgerichtet ist und ansonsten keine weiteren Funktionalitäten bietet. Der Fokus liegt hierbei auf englischsprachigen Texten aus Social-Media-Plattformen. Es wurde im Jahr 2014 von C.J. Hutto als Open-Source-Projekt unter der MIT Lizenz auf Github veröffentlicht und hat seitdem einige Verbesserungen durch Mitwirkende erfahren. Zur Sentiment-Analyse nutzt VADER eine Kombination aus einem Lexikon-basierten Ansatz und einem auf grammatikalischen und syntaktischen Regeln basierendem Ansatz.(vgl. Hutto und Gilbert 2015, S. 1) Das Lexikon wurde auf Basis bestehender Sentiment-Lexika erstellt. Hinzu kommen Text-basierte Emoticons (Bspw. „:/“ „;)“), auf Social-Media-Plattformen häufig genutzte Idiome

und Akronyme (lol, wtf, rofl), welche häufig Rückschlüsse auf das zugrundeliegende Sentiment eines Textes zulassen können. Die Einträge, die aus der Kombination der o.g. Lexika gewonnen werden konnten, seien dann von 10 vorab geprüften und trainierten Nutzern auf AMT² mit Werten von -4 (extrem negativ) bis +4 (extrem positiv) annotiert worden. Nach statistischer Filtrierung seien >7500 Einträge mit einem von Null abweichenden, negativen oder positiven Sentiment-Wert übrig geblieben. (vgl. Hutto und Gilbert 2015, S.5) Diese Anzahl wurde später noch durch die Unterstützung von UTF-8 kodierten Emojis³ erweitert. Die Autoren führen fünf durch qualitative Analyse bestimmte Heuristiken auf, welche die Intensität des Sentiments beeinflussen sollen:

1. **Interpunktion:** Das Ausrufezeichen erhöhe die Intensität, ohne die Orientierung des Sentiments zu verändern.
2. **Majuskelgebrauch:** Wird ein für das Sentiment relevantes Wort, unter Präsenz anderer kleingeschriebener Wörter, ausschließlich in Großbuchstaben geschrieben, so erhöhe dies die Intensität des Sentiments, ohne dabei die Orientierung des Sentiments zu verändern.
3. **Modaladverbien:** Diese können ihnen folgende Adjektive graduieren und hätten somit Einfluss auf die Intensität des geäußerten Sentiments.
4. **Konjunktionen (in kontrastiver Bedeutung):** Das Wort aber signalisiere einen Wechsel der Sentiment-Polarität. Dabei sei das Sentiment des Teils des Satzes der dem „aber“ folgt der dominanter Teil.
5. **Negationen:** Durch Betrachtung der Trigramme, welche einem für das Sentiment relevantem Wort vorausgehen, sei es möglich einen Großteil der Fälle, in denen die Polarität des Satzes durch Negationen umgedreht wird, zu erkennen.

(Hutto und Gilbert 2015, S.6, Abs. 3.2, übersetzt MF, Beispiele ausgelassen.)

5.6.2. TextBlob

TextBlob ist eine Python-Bibliothek für die Verarbeitung von Texten. Sie wird seit 2013 als Open-Source-Projekt, unter der MIT Lizenz auf Github entwickelt. Der Code baut zum Teil auf anderen Open-Source-Projekten auf, so zum Beispiel auch auf dem Natural-Language-Toolkit(NLTK)⁴ und der pattern-Bibliothek⁵. TextBlob bietet für die Sentiment-Analyse zwei Lösungen mit jeweils verschiedenen theoretischen Ansätzen an: den NaiveBayesAnalyser und den PatternAnalyser.

NaiveBayesAnalyser Der naive Bayes Klassifikator basiert auf der gleichnamigen Implementation in dem natural Language Toolkit. Die theoretische Grundlage des Klassifikators ist im wesentlichen äquivalent zu dem was in Abs. 2.1.2.1 vorgestellt wurde. Das Training dieses Klassifizierers läuft hierbei mit einem in NLTK enthaltenen Korpus. Der Datensatz „movie_reviews“ enthält Filmrezensionen in englischer Sprache, welche mit den Labeln „positive“ oder „negative“ annotiert wurden.

²<https://www.mturk.com/>, letzter Zugriff 15.08.2021.

³<https://unicode.org/emoji/charts/full-emoji-list.html>, letzter Zugriff: 15.08.2021.

⁴<https://www.nltk.org/>, letzter Zugriff: 21.08.2021.

⁵<https://github.com/clips/pattern>, letzter Zugriff: 21.08.2021.

PatternAnalyzer Die PatternAnalyzer Implementation ist eine Erweiterung der Klasse „sentiment“ aus der pattern-Bibliothek (Smedt und Daelemans 2012). Diese Bibliothek nutzt einen Parser, welchem ein Lexikon und Heuristiken zugrunde liegen. Mit diesem Parser können für Sätze Token, POS-Tags und die Rolle der jeweiligen Wörter bestimmt werden. Des Weiteren enthält die pattern Bibliothek eine Liste mit Adjektiven. Jedes dieser Adjektive wurde mit einem Sentiment-Wert im Wertebereich von -1(negativ) bis +1(positiv) versehen. Die Sentiment-Analyse besteht nun darin, den Mittelwert aller Sentiment-Werte der im Satz enthaltenen Adjektive zu bestimmen. Interessanterweise liefert diese Implementation auf analoge Weise auch einen so genannten subjectivity score, welcher einen Wertebereich von 0(objektiv) bis 1(subjektiv) hat.

5.6.3. Stanza

Bei Stanza (Qi u. a. 2020) handelt es sich wie bei den anderen, in dieser Arbeit verwendeten Werkzeugen, um ein Open-Source Python Modul. Dieses wurde von der Stanford NLP Gruppe unter einer Apache 2.0 Lizenz veröffentlicht. Des Weiteren integriert das Modul auch eine API für die Stanford Core-NLP Bibliothek (Manning u. a. 2014), welche zuvor nur für die Programmiersprache Java verfügbar war. Stanza beinhaltet Werkzeuge für Standard NLP-Aufgaben, wie Tokenisierung, POS-Tagging, Morphological Feature-Tagging, dependency parsing, named entity recognition und Sentiment-Analyse. In dieser Arbeit findet allerdings nur das Werkzeug zur Sentiment-Analyse Anwendung. Diese ist derzeit (Stand August 2021) für die Sprachen Englisch, Chinesisch und Deutsch verfügbar bzw. vortrainiert. Das dem Sentiment-Klassifizierer zugrundeliegende Modell ist ein convolutional neural network nach Kim (2014). Als Input nutzt dieses CNN die im Grundlagen-Kapitel vorgestellten word2vec-Wort-Vektoren nach Mikolov, Chen u. a. (2013). Auf die Eingabe von Wort-Vektoren folgt ein Convolutional-Layer(wobei mehrere Feature-Kanäle berücksichtigt werden), ein Max-Pooling-Layer und ein Feed-Forward-NN mit Softmax-Aktivierungsfunktion im Ausgabe-Layer. Das NN nutzt zusätzlich noch einen Dropout, um ein Overfitting auf die Trainingsdaten zu vermeiden.

Kim gibt an, dass das CNN mit der beschriebenen Struktur auf verschiedenen Testdatensätzen vergleichbar mit Deep-Learning-Modellen abschneidet. Das vortrainierte Modell, welches in der Stanza Bibliothek mitgeliefert wird, sei mit der Stanford Sentiment Treebank(+zusätzlichen Sätzen) und weiteren Datensätzen trainiert worden. Es erreicht laut den Autoren der Dokumentation⁶ mit einer Klassifikation in 3 Klassen $\in \{\text{Positiv, Negativ, Neutral}\}$, trainiert mit den o.g. Datensätzen, auf dem SST Test-Datensatz eine Accuracy von 70%.

5.6.4. Huggingface

Die Transformers Bibliothek bietet die Möglichkeit verschiedene Transformer-Modelle über eine standardisierte Schnittstelle zu verwenden. Dabei ist die Verwendung der Modelle sowohl über das PyTorch-Backend, als auch über das TensorFlow-Backend möglich. Darüber hinaus sind zahlreiche von der Community für verschiedene NLP-Aufgaben fein-abgestimmte

⁶<https://stanfordnlp.github.io/stanza/sentiment.html>, letzter Zugriff: 30.08.2021.

Modelle verfügbar. In dieser Arbeit werden für die Sentiment-Analyse die folgenden auf BERT (Devlin u. a. 2019) basierenden Modelle betrachtet:

distilBERT Die distilBERT-Architektur soll laut den Autoren durch einen angepassten Trainingsprozess, im Vergleich zu BERT, bei einer Reduktion der Größe um 40%, 60% schneller sein und 97% der Fähigkeit zum Verstehen von natürlichen Sprachen behalten. (Sanh u. a. 2020) Das Training für die Sentiment-Analyse wurde mit dem SST2-Datensatz (Stanford-Sentiment-Treebank (Socher u. a. 2013)) durchgeführt.

BERTweet Dieses Modell basiert auf der $BERT_{base}$ -Architektur und wurde analog zu RoBERTa (Y. Liu u. a. 2019) auf einem Datensatz von 850 Millionen Tweets vortrainiert. Für die Sentiment-Analyse wurde der Datensatz aus SemEval2017 Task 4A, Rosenthal, Farra und Nakov (2017) verwendet. Nach Angaben der Autoren erzielt das so konstruierte Modell, u.a. bei Tweet-Text-Klassifikation, bessere Ergebnisse als die Modelle RoBERTa_{base} und XLM-R_{base}. (vgl. D. Q. Nguyen, Vu und A. T. Nguyen 2020, S.1)

finBERT Dieses Modell wurde mit einem Korpus aus der Finanzdomäne (TRC2⁷) vortrainiert. Die Sentiment-Analyse wurde mit englischen Sätzen aus dem Financial-Phrasebank-Datensatz, Malo u. a. (2013) trainiert. Nach Angaben des Autors wurde hiermit eine Accuracy von 97% erzielt. (vgl. Araci 2019, S. 6)

5.6.5. fastText

Nach Anregung von G. Abrami wurde ein Klassifizierer basierend auf Fasttext-Embeddings trainiert. Zum Training eines solchen Supervised-Modells wird ein Datensatz mit großer Anzahl von gelabelten Sätzen benötigt. Dieser wurde aufgrund des großen Aufwands für die Annotation von genügend Tweets, abweichend zum Vorgehen beim Testdatensatz, nicht manuell erstellt. Stattdessen wurde eine Teilmenge von 100.000 zufällig ausgewählten normalen Tweets, welche keine URL enthalten, geladen, in Sätze zerlegt, vorverarbeitet und von allen implementierten Klassifizierern (TextBlob:PatternAnalyzer, TextBlob:NaiveBayesAnalyzer, VADER, Stanza, distilBERT, BERTweet, finBERT), wie in Abs. 4.2 beschrieben klassifiziert. Danach wurden Sätze mit einer Übereinstimmung von 6 Klassifizierern behalten und in einen Trainings- und Validierungsdatensatz aufgespalten. Das Modell nach Joulin u. a. (2016) ist eine Variation der im Grundlagen-Kapitel vorgestellten CBOW-Architektur nach Mikolov, Chen u. a. (2013). Das Wort dessen Vorhersage trainiert wird ist hierbei das Klassen-Label, also für die betrachtete binäre Sentiment-Klassifizierung {Positive, Negative}. (Joulin u. a. 2016) Für das auf den Daten vortrainierte Modell wurde anschließend noch eine Wrapper-Klasse geschrieben, um es analog zu den anderen Modellen verwenden zu können.

⁷Reuters TRC2, <https://trc.nist.gov/data/reuters/reuters.html>, zuletzt aufgerufen am: 08.09.2021.

5.7. Performanz

Umstellung der Transformer auf PyTorch-CUDA-Implementierung Die Transformer Modelle aus der Huggingface Bibliothek nutzen standardmäßig keine Beschleunigung über CUDA-fähige Grafikkarten (Stand August 2021). Aus diesem Grund wurden für die Modelle anstatt der vorgefertigen Pipeline eigene Wrapper Klassen geschrieben. Diese spalten sich auf in zwei Teile: den Tokenizer und das Modell (inklusive Feed-Forward-Netzwerk am Ende mit Softmax-Aktivierungsfunktion). Beim Tokenizer wurde außerdem jeweils, falls vorhanden, eine schnellere Implementierung in Rust vorgezogen. Die Klasse wurde dann jeweils durch den höchsten Wert im Ausgabe-Layer bestimmt.

Multiprocessing Da es sich weiterhin um einen sehr großen Datensatz handelt, wurde außerdem ein Multiprocessing-Wrapper geschrieben, welcher ermöglicht, dass die Tweets parallel verarbeitet werden können. Da zwischen den einzelnen Klassifikationen keine Abhängigkeiten bestehen, wurde der Pooling-Ansatz der Python Multiprocessing Bibliothek gewählt.

Tabelle 5.2.: Vergleich der durchschnittlichen Inferenz-Zeiten der verschiedenen Sentiment-Analyse Lösungen

| Sentiment-Analyzer | <i>durchschnitt. Zeit für je 10.000 zufällig ausgewählte Tweets</i> |
|------------------------------|---|
| TextBlob: PatternAnalyzer | 14 Sekunden |
| TextBlob: NaiveBayesAnalyzer | 51 Sekunden |
| VADER | 7 Sekunden |
| Stanza | 495 Sekunden |
| distilBert | 710 Sekunden |
| BERTweet | 1340 Sekunden |
| FinBERT | 1355 Sekunden |
| fastText | 7 Sekunden |
| MajorityVote | 5950 Sekunden |

6. Darstellung der Ergebnisse

6.1. Evaluation der Sentiment-Klassifizierer

In A.1 werden die Ergebnisse der Evaluation der verschiedenen Sentiment-Analyse-Lösungen aufgeführt. Dabei wurden, wie in Abs. 4.2 beschrieben, „harte“ Klassenzuweisungen bestimmt und bzgl. des Testdatensatzes, wurde wie in den Abschnitten 4.2, 5.5 beschrieben, vorgegangen. Dabei hat sich gezeigt, dass dabei der MajorityVote-Ansatz mit einer Accuracy von 0.80 das beste Ergebnis erzielt hat. Am schlechtesten hat das distilBERT-Modell abgeschnitten. Beachtet werden sollte hierbei, dass für die Klassifikation in drei Klassen eine niedrigere Baseline für die Accuracy angesetzt werden sollte (0.333 mit uniform zufälliger Auswahl der Klassen, verglichen mit 0.5 bei binärer Klassifikation).

6.2. Zusammenhang der Zeitreihen

6.2.1. Allgemein

Der Abb. A.1 lassen sich die Distanzkorrelationen der aggregierten Sentiment-Zeitreihen und der Bitcoin-Rendite-Zeitreihe entnehmen. Dabei ist der höchste Wert für die Distanzkorrelation bei finBERT mit 0.56 bei den auf Tagesbasis aggregierten Sentiment-Werten festzustellen. Der höchste Wert für die Distanzkorrelation für die auf Stundenbasis aggregierten Sentiment-Werten ist ebenfalls bei finBERT mit 0.16 zu finden. Dabei ist insgesamt festzuhalten, dass die auf Stundenbasis aggregierten Sentiment-Werte geringere Werte für die Distanzkorrelationen aufweisen als die auf Tagesbasis aggregierten Sentiment-Werte.

Die Pearson-Korrelationen lassen sich Abb. A.2 entnehmen. Hierbei weisen finBERT für die auf Tagesbasis aggregierten Sentiment-Werte den höchsten Wert auf. Auf die fortfolgende Betrachtung der auf Stundenbasis aggregierten Sentiment-Zeitreihen wird aufgrund der Beobachtung derselben Tendenzen, aber bezüglich niedrigeren Baseline-Werte und in Hinblick auf den Umfang der Arbeit verzichtet¹.

6.2.2. Filtration mittels Tweet-basierten Merkmalen

In Abb. A.3 werden die Distanzkorrelationen mit den von Tweet-basierten Merkmalen induzierten Teilmengen aufgeführt. Dabei stellt die oberste Zeile in den Abbildungen immer die Baseline dar und die darauffolgenden Zeilen die Distanzkorrelationen bei den durch das jeweilig angegebene Merkmal induzierten Teilmengen.

Die Betrachtungen, ob ein Merkmal insgesamt einen nennenswerten Einfluss hat, wurden über die Zeilendurchschnitte getroffen (Baseline-Zeilendurchschnitt von 0.375).

Merkmal URL Für das Merkmal, dass ein Tweet eine URL enthält wurde ein leicht zunehmender Zeilendurchschnitt (0.381) im Vergleich zur Baseline gefunden.

Für das Merkmal, dass ein Tweet keine URL enthält wurde hingegen ein leicht abnehmender Zeilendurchschnitt (0.365) im Vergleich zur Baseline gefunden.

Merkmal free Für das Merkmal ob der Tweet das Wort „free“ oder eine Variation enthält wurde ein Abnahme des Zeilendurchschnitts (0.352) gefunden.

Für das gegenteilige Merkmal, dass der Tweet keine Variation des Wortes „free“ enthält, konnte eine Zunahme des Zeilendurchschnitts (0.382) gefunden werden.

Merkmal Tweetart Für das Merkmal, dass es sich um einen normalen Tweet handelt konnte eine starke Zunahme des Zeilendurchschnitts (0.446) festgestellt werden.

Für das Merkmal, dass es sich um Retweets handelt konnte eine starke Abnahme des Zeilendurchschnitts (0.343) gefunden werden.

Für die Merkmale, dass es sich um Reply und Quote handelt konnten auch abnehmende Zeilendurchschnitte (0.368 u. 0.365) festgestellt werden.

¹Sie sind der Vollständigkeit halber dennoch im Zubehör dieser Arbeit enthalten und können bei Interesse in dem Ordner „results“ gefunden werden.

6.2.3. Filtration mittels User-basierten Merkmalen

Die Darstellungen sind analog zu den vorherigen zu interpretieren und die Betrachtungen wurde analog anhand des Zeilendurchschnitts getroffen.

Verifizierung Für das Merkmal ob der Autor eines Tweets von Twitter verifiziert wurde, wurde eine relativ starke Zunahme des Zeilendurchschnitts (0.418) festgestellt.

Followeranzahl Für das Merkmal der Followeranzahl des Autors, konnte mit der Followerzahl zunehmende Zeilendurchschnitte gefunden werden, dieser Trend kehrt allerdings bei 100.000 Followern (0.496) um und die Zeilendurchschnittswerte nehmen wieder ab.

Followeeanzahl Für das Merkmal der Followeeanzahl bzw. der Anzahl von Nutzern denen der Autor selbst folgt, konnten sich keine nennenswerten Muster erkennen lassen.

Zeitdelta Bei dem Merkmal der Zeitdauer zwischen Erstellung des Accounts und des Posten des Tweets, konnte ein leicht zunehmender Trend (0.396 bei 10 Jahren) beobachtet werden.

Gesamte Anzahl Likes & Tweets Die beiden Merkmale der Anzahl von Tweets des Nutzers und die Anzahl der Likes des Nutzers wiesen auch leicht zunehmende Trends auf.

6.2.4. Kombination der Merkmale

Für jeden der neun in dieser Arbeit betrachteten Klassifizierer werden jeweils die drei höchsten Werte der Distanzkorrelation zwischen den aggregierten Sentiment-Zeitreihen der durch die Kombination der Merkmalen induzierten Teilmengen und den Bitcoin-Rendite-Zeitreihen in der Tab. A.2 aufgeführt. Dabei wurden Teilmengen, die durch die Kombination der Merkmale nur noch weniger als 1000 Repräsentanten enthielten, verworfen. Die Anzahlen der Elemente in den Teilmengen lassen sich Abb. A.5 entnehmen. Die fehlenden Werte, welche mit "N/A" gekennzeichnet wurden, resultieren wie in Abschnitt 4.3 beschrieben daraus, dass Merkmale derselben Art nicht kombinierbar sind. Wie der Tabelle zu entnehmen ist, findet sich der größte somit zu berücksichtigende Wert bei dem Klassifizierer finBERT mit einer Distanzkorrelation von 0.718 bei der Teilmenge, bestehend aus normalen Tweets, welche keine URL enthalten.

7. Diskussion der Ergebnisse

7.1. Sentiment-Klassifizierer

7.1.1. TextBlob: NaiveBayesAnalyzer

Der NaiveBayesAnalyzer wurde nur mit Filmrezensionen trainiert. Dies hat zur Folge, dass diesem wohl einige Wahrscheinlichkeiten für Token fehlen werden und somit nicht berücksichtigt werden können. Zudem kann ein naiver Bayes Klassifikator begründet durch die Kommutativität des multiplikativen Ansatzes keinen Kontext wahrnehmen. Damit folgt u.A., dass die Reihenfolge der Wörter in den Sätzen keine Rolle spielt. Zugutehalten lässt sich dem Klassifizierer, dass die Inferenz für einen „lern-basierten“ Ansatz im Vergleich sehr schnell ist.

7.1.2. TextBlob: PatternAnalyzer

Der Ansatz des PatternAnalyzers ist die Nutzung eines Parsers zur syntaktischen Analyse der Sätze und die darauffolgende Zuweisung eines Sentiment-Werts anhand einer bewerteten Liste von Adjektiven. Ein direkt erkennbares Problem hierbei ist, dass auf Twitter häufig auch verkürzte Sätze, ohne valide Grammatik auftreten. Ein weiteres denkbare Problem ist, dass möglicherweise nicht-enhaltene oder aber anders konnotierte Adjektive in der Domäne vorkommen. Allerdings scheint das oben Genannte keinen großen Einfluss gehabt zu haben, denn nichtdestotrotz schneidet der Klassifizierer mit einer Accuracy von 68% relativ gut ab und die Inferenz ist erwartungsgemäß sehr schnell.

7.1.3. Stanza

Das vortrainierte Stanza-CNN Modell wurde auf mehreren Datensätze aus verschiedenen Domänen trainiert. Das Modell zeigt mit einer Accuracy von 73% eine gute Fähigkeit zur Generalisierung. Aufgrund des neuronalen Ansatzes dauert die Inferenz jedes einzelnen Sentiment allerdings deutlich länger als bei den Regel- und Lexikon-basierten Klassifizierer und wird nur noch von den Transformer-basierten Modellen übertroffen.

7.1.4. VADER

VADER wurde speziell für englische Tweets entwickelt und kann Emojis, Redewendungen, wie sie auf Twitter verwendet werden, in den Analysen zum Sentiment berücksichtigen. Allerdings wurde hierbei nur eine gering bessere Accuracy im Vergleich zu dem TextBlob: PatternAnalyzer festgestellt, was darauf hindeutet, dass viel der Information welches Sentiment ein Tweet ausdrücken soll bereits über die verwendeten Adjektive abgedeckt wird. Erwähnenswert ist dennoch, dass die Inferenzen mit dem Ansatz sehr schnell sind (ca. 7 Sekunden je 10.000 Tweets).

7.1.5. distilBERT

Das distilBERT-Modell wurde auf SST-Datensatz vortrainiert. Der Ansatz schneidet mit einer Accuracy von 0.44 am schlechtesten ab. Das Problem hierbei ist es, dass das Modell nur eine binäre Klassifikation ermöglicht und der Ansatz zur Mitigation dessen (vgl. hierzu Abschnitt 4.2), wie sich in dem F1-Score für die Klasse „Neutral“ von 0.14 widerspiegelt, nur bedingt hilfreich ist. Die Inferenz dieses Modells ist allerdings, vermutlich wegen der reduzierten Komplexität, deutlich schneller als die anderen betrachteten BERT-basierten Modelle und mit ca. 12 Minuten je 10.000 Tweets nur ungefähr 4 Minuten langsamer als Stanza.

7.1.6. BERTweet

Der Encoder des BERTweet Modells wurde mit einem großen Korpus von Tweets vortrainiert. Außerdem nutzt dieser einen abweichenden Tokenizer-Ansatz, bei dem im Gegensatz zu den anderen in dieser Arbeit betrachteten BERT-basierten Modellen, URLs und Emojis berücksichtigt werden können. Dieser Ansatz hat mit einer Accuracy von 79% am besten abgeschnitten. Die Inferenz ist allerdings, begründet durch die Komplexität des Modells, mit ca. 22 Minuten je 10.000 Tweets relativ langsam.

7.1.7. finBERT

Beim finBERT-Modell wurde sowohl das Encoding-Layer mit einem Korpus bestehend aus Wörtern mit Finanzbranchen-Bezug vortrainiert, als auch die Sentiment-Analyse mit einem Datensatz aus der Finanzdomäne trainiert. Hierbei ist festzuhalten, dass der Korpus aus Reuters Artikeln besteht und daher wohl eher eine förmlichere Sprache beinhaltet, als sie hauptsächlich auf Twitter vorzufinden ist. Dies gilt wahrscheinlich auch für den Datensatz, welcher zum Training der Sentiment-Klassifikation genutzt wurde. Außerdem ist festzustellen, dass obwohl die Terminologie beim Handel von Kryptowährungen sehr viel Überschneidung mit der Terminologie beim Handel von traditionellen Anlageklassen hat, einige elementare Wörter und Phrasen wohl nicht im Korpus und im Trainingdatensatz vorhanden sind. Die Inferenz von 10.000 Tweets dauert ungefähr genau so lange wie bei BERTweet, was bei der Betrachtung der sehr ähnlichen Architekturen allerdings nicht verwundern sollte.

7.1.8. fastText Klassifizierer

Das Modell wurde auf einem Teil des Twitter-Datensatzes trainiert. Dabei wurden die Labels zum Training, wie in Abs. 5.6.5 beschrieben, generiert. Dadurch konnte ein relativ großer Domänen-spezifischer gelabelter Datensatz generiert werden. Die Methodik hat allerdings den Nachteil, dass zwischen Klassifizierer nur bei positiven und negativen Sätzen genügend Einigkeit gegeben ist und deshalb nur eine binäre Klassifikation trainiert werden konnte. Dadurch treten ähnliche Probleme wie bei distilBERT auf, wodurch mögliche Vorteile durch das Training auf dem domänenspezifischen Datensatz nicht zum Tragen kamen.

7.1.9. Majority-Vote Klassifizierer

Der konstruierte MajorityVote Klassifizierer, welchem die Zuweisungen aller anderen Klassifizierer zugrundeliegen, schneidet mit einer Accuracy von 80% am besten ab. Da hierfür aber jeweils alle anderen Klassifizierer durchlaufen müssen, ist jeder Inferenzschritt relativ zeitaufwendig und er braucht damit 100 Minuten für 10.000 Tweets. Die Ursache der Tatsache, dass die Inferenz des MajorityVote Klassifizierers deutlich länger dauert, als die Summe der durchschnittlichen Einzeldauern der Klassifizierer, wird in Limitationen des GPU-Speichers und der Bandbreite des PCI-Bus gesehen, welche bei gleichzeitiger Verwendung der Modelle in Bezug auf die Dauer negativ ins Gewicht fallen könnten.

7.2. Zusammenhang der Zeitreihen

7.2.1. Allgemein

Für alle Klassifizierer lassen sich zwischen den aggregierten Sentiment-Zeitreihen und den Bitcoin-Rendite-Zeitreihen Distanzkorrelationen ungleich 0 feststellen. Für die auf Stundenbasis aggregierten Sentiment-Werte sind die Werte für die Distanzkorrelation geringer als für die auf Tagesbasis aggregierten Sentiment-Werte, was dafür spricht, dass der Zusammenhang verzögert ist.

7.2.2. Tweet-basierten Merkmalen

Anders als erwartet wurde ein zunehmender Zusammenhang mit den Sentiments von Tweets die eine URL enthalten gefunden und ein abnehmender für Tweets die keine URL enthielten. Dies könnte zum Beispiel daran liegen, dass anders als über die kurze Einsichtnahme vermutet, Tweets die URLs enthalten, überwiegend auf Nachrichten-Artikel verweisen, welche die Kaufentscheidungen der Lesenden beeinflussen (z.B. durch stark polarisierende Headlines oder die enthaltenen Informationen). Die Feststellungen bezüglich der anderen Merkmale decken sich mit den in Tabelle 4.2 formulierten Annahmen und werden an dieser Stelle nicht noch einmal wiederholt.

7.2.3. User-basierten Merkmalen

Hierzu konnte festgestellt werden, dass anders als erwartet, die Merkmale der gesamten Anzahl von Likes und Tweets, die ein Autor getätigt hat, einen Einfluss haben. Der Zuwachs an beobachteten Zusammenhang ist allerdings sehr gering. Als möglicher Grund für den Zuwachs mit der Anzahl von Tweets könnte die Erhöhung der Wahrscheinlichkeit dienen, dass die in Tweets geäußerten Meinungen von anderen Nutzern eher gesehen werden, wenn sie denn nur oft genug wiederholt werden. Die Erkenntnisse zu den anderen Merkmalen decken sich hier auch mit denen in Tabelle 4.3 formulierten Annahmen.

7.2.4. Kombination der Merkmale

Die Kombination der Merkmale hat nicht zu einem besonderen Zuwachs an Erkenntnissen geführt. Es hat sich dabei gezeigt, dass auch hier vor allem die Merkmale der Followeranzahl (insbesondere ≥ 100.000 Follower) und das es sich um einen originären, „normalen“ Tweet handelt, die größten Auswirkungen auf den Zusammenhang haben.

Die Feststellung, dass bei dem NaiveBayesAnalyzer und distilBERT abweichend hierzu, Kombinationen mit dem Merkmal, dass ein Tweet eine URL enthält, zu dem größten Zugewinn von Zusammenhang führen, wird durch die Erkenntnisse der Evaluation abgeschwächt.

7.3. Beschränkungen

- Der betrachtete Zeitraum umfasst sowohl einen Teil in dem die Kurse über mehrere Tage hinweg steigen als auch fallen. Es ist z.B. möglich, dass in einem Zeitraum in dem über längere Zeit hinweg nur eine Seitwärts-Bewegung des Kurses vorhanden ist, die in der Gleichung 4.1 definierten aggregierten Sentiment-Werte an Aussagekraft verlieren, falls dadurch insgesamt weniger Tweets oder stark überwiegend neutrale Tweets vorkommen.
- Bei dem fastText-Klassifizierer wurde beim Erstellen des Trainingsdatensatzes nicht darauf geachtet, ob es möglicherweise Überschneidungen mit dem Testdatensatz gibt. Dabei wären nur die Texte selbst betroffen, allerdings nicht die angefertigten Annotationen. Da das fastText Modell sowieso eher schlechte Ergebnisse erzielt, sind die Auswirkungen auf die Aussagekraft der Ergebnisse dieser Arbeit als sehr gering einzuschätzen.
- Die Beibehaltung des Namens „MajorityVote“ für die Zuweisung eines weichen Sentiment-Werts ist aus denen in Abschnitt 4.3 genannten Gründen problematisch. Es ist aufgrund der Evaluationsergebnisse anzunehmen, dass der dort beschriebene Ansatz einen höheren Zusammenhang als die Vereinfachung aufweisen würde.
- Es ist weiterhin auch möglich, dass die in dieser Arbeit festgestellten Effekte zum Teil auch durch nicht beobachtete Variablen beeinflusst wurden.

8. Ausblick und Fazit

8.1. Mögliche Erweiterungen

Ausweitung auf weitere Handelspaare In dieser Arbeit wurde nur das vom Handelsvolumen wichtigste Handelspaar Bitcoin/USDT betrachtet. Eine sehr einfache Erweiterung wäre die Untersuchung weiterer Handelspaare. Ein geeigneter Vertreter wäre hierbei wohl die in Bezug auf Handelsvolumen auf Bitcoin folgenden Kryptowährung Ethereum. Falls das Volumen der Tweets ausreichend ist und die Analyse für Ethereum ähnliche Schlüsse ziehen lässt wäre eine Berücksichtigung des Sentiment-Werts in der conservation function bei AMMs (automated market makers) auf dezentralisierten Handelsplätzen¹ zur Prävention von Zusammenbrüchen von „Stable Coins“, wie sie in der Vergangenheit stattgefunden haben, eine gute Idee. Außerdem könnte der Sentiment-Wert in der Praxis dann auch hervorragend zum Erkennen von Liquidität beim Handel von über NFTs (Non fungible Tokens)² abgebildeten Assets, wie Kunstrechten, Domains, etc. eingesetzt werden.

Berücksichtigung von Follower-Netzwerken Es wurde weiterhin überlegt Follower-Netzwerke als weiteres Feature miteinzubeziehen. Damit wäre es zum Beispiel möglich, die sich als höchst relevantes Merkmal herausgestellte Anzahl der Follower weiter zu differenzieren. So ist es zum Beispiel denkbar, dass ein Tweet von Elon Musk, über Veränderungen in der Transaktions-Geschwindigkeit des Dogecoins einen höheren Einfluss auf die Preisbildung hat, als ein Tweet von jemandem, der sonst nichts mit Kryptowährungen zu tun hat.(vgl. hierzu Ante (2021))

Differenzierung nach Emotionen Bei der Recherche nach passenden Sentiment-Klassifikations-Modellen, wurde auch ein Transformer-Modell gefunden, welches einem Text die zugrundeliegende Emotion zuordnen können sollte. (Huang u. a. 2019, vgl.) Vorbehaltlich das dies auf Twitter-Daten entsprechend gut funktioniert, könnten die in dieser Arbeit vorgeschlagenen Experimente in abgewandelter Form wiederholt werden und nach den jeweiligen Emotionen differenziert betrachtet werden.

Ausweitung auf andere Kontexte Eine weiterführende Arbeit könnte auch die Untersuchungen auf andere Kontexte ausweiten. Ein beispielhafter Sachverhalt mit aktueller Relevanz wäre der Deutsche Wahlkampf 2021. Dabei könnte zum Beispiel betrachtet werden wie sich die aggregierten Sentiment-Werte rund um TV-Auftritte etc. verändern und wie sich diese zu den über Telefon-Abfragen erhobenen Zustimmungswerten verhalten.

¹Siehe hierzu (Xu u. a. 2021).

²Siehe hierzu: <https://opensea.io/>, letzter Aufruf: 18.09.2021.

Weiterentwicklung des MajorityVoting-Ansatzes Außerdem könnten weiterführende Arbeiten den Ansatz des Meta-Modells für Sentiment-Klassifikation weiter ausbauen und optimieren. Zum einen wäre es vermutlich möglich, die Inferenzdauer zu senken, wenn die Berechnungen der einzelnen Klassifizierer so umgestaltet werden, dass sie asynchron ablaufen. Zum anderen wäre hier auch ein lernbasiertes Meta-Modell ähnlich dem Ansatz von Hemati und Mehler (2019) denkbar.

8.2. Fazit

Ziel dieser Arbeit war die Untersuchung von Sentiment-Zeitreihen von Tweets auf Zusammenhang mit den Kursverläufen der Kryptowährung Bitcoin.

Dazu wurden zuerst verschiedene Ansätze zur Sentiment-Analyse auf ihre Tauglichkeit in der Domäne untersucht. Dabei haben sich vor allem ein auf BERT basierendes Modell und das konstruierte Meta-Modell als sehr vielversprechend herausgestellt.

Die Untersuchungen des Zusammenhangs, gemessen durch Distanzkorrelation zwischen den aggregierten Sentiment-Werten und den Renditen des Bitcoins, haben ergeben, dass dieser für finBERT, ein speziell für Finanzmarktkontexte angepasstes BERT-Modell, am höchsten ist.

Die Betrachtungen hinsichtlich des Einflusses der Merkmale von Tweets haben gezeigt, dass insbesondere originäre Tweets und Tweets von Nutzern mit hoher Followeranzahl die größten Auswirkungen auf den Zusammenhang haben.

Die in dieser Arbeit erzielten Erkenntnisse über das Verhältnis des Sentiment-Werts und der Kurse der Kryptowährung Bitcoin eröffnen eine Vielzahl von Anwendungsmöglichkeiten. Diese reichen von dem Ausnutzen eines Vorteils über den Markt zum Zwecke der persönlichen Profitsteigerung bis hin zur Prävention von Totalausfällen für Anleger.

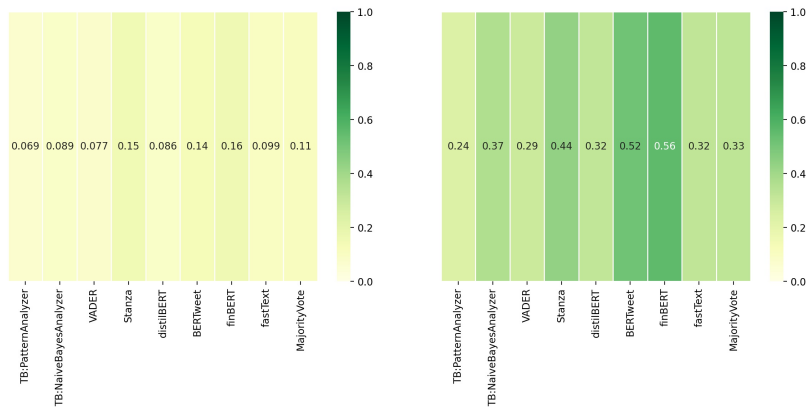
A. Ausführliche Ergebnisse

A.1. Evaluation der Sentiment-Klassifizierer

Tabelle A.1.: Vergleich Evaluationsmaße der Klassifizierer auf dem erstellten Testset (insg. 500 Tweets, davon 129 negativ; 215 neutral; 156 positiv).

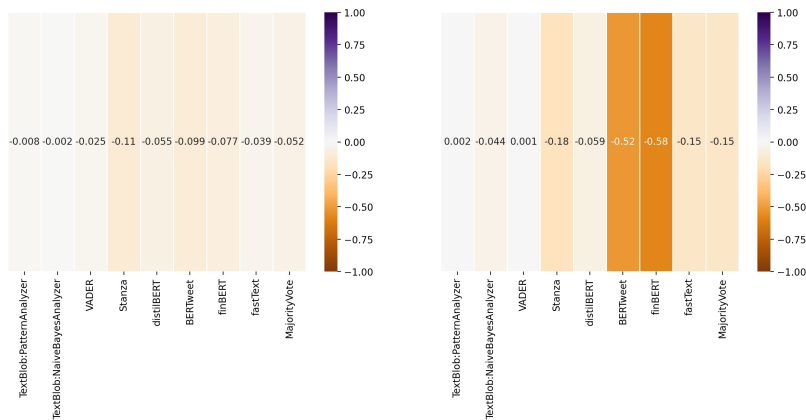
| Sentiment-Klassifizierer | Klasse | <i>Precision</i> | Recall | F1-Score | Makro-F1-Score | Accuracy |
|--------------------------|---------|------------------|--------|----------|----------------|----------|
| TB: PatternAnalyzer | Negativ | 0.83 | 0.71 | 0.76 | 0.69 | 0.68 |
| | Neutral | 0.70 | 0.59 | 0.64 | | |
| | Positiv | 0.59 | 0.80 | 0.68 | | |
| TB: NaiveBayesAnalyzer | Negativ | 0.54 | 0.38 | 0.45 | 0.46 | 0.47 |
| | Neutral | 0.46 | 0.52 | 0.49 | | |
| | Positiv | 0.44 | 0.48 | 0.46 | | |
| VADER | Negativ | 0.73 | 0.79 | 0.76 | 0.71 | 0.70 |
| | Neutral | 0.79 | 0.57 | 0.66 | | |
| | Positiv | 0.61 | 0.80 | 0.69 | | |
| Stanza | Negativ | 0.71 | 0.67 | 0.69 | 0.71 | 0.73 |
| | Neutral | 0.67 | 0.92 | 0.78 | | |
| | Positiv | 0.95 | 0.52 | 0.67 | | |
| distilBERT | Negativ | 0.34 | 0.90 | 0.49 | 0.43 | 0.44 |
| | Neutral | 0.37 | 0.08 | 0.14 | | |
| | Positiv | 0.82 | 0.56 | 0.66 | | |
| BERTweet | Negativ | 0.83 | 0.80 | 0.81 | 0.79 | 0.78 |
| | Neutral | 0.74 | 0.81 | 0.77 | | |
| | Positiv | 0.81 | 0.73 | 0.77 | | |
| finBERT | Negativ | 0.93 | 0.49 | 0.64 | 0.65 | 0.67 |
| | Neutral | 0.58 | 0.96 | 0.72 | | |
| | Positiv | 0.92 | 0.43 | 0.59 | | |
| fastText | Negativ | 0.44 | 0.91 | 0.60 | 0.53 | 0.55 |
| | Neutral | 0.75 | 0.18 | 0.29 | | |
| | Positiv | 0.65 | 0.78 | 0.71 | | |
| MajorityVote | Negativ | 0.87 | 0.81 | 0.84 | 0.80 | 0.80 |
| | Neutral | 0.76 | 0.82 | 0.79 | | |
| | Positiv | 0.81 | 0.76 | 0.79 | | |

A.2. Experiment Teil 1



(a) Auf Stundenbasis aggregiert. (b) Auf Tagesbasis aggregiert.

Abbildung A.1.: Distanzkorrelationen der auf Stunden- und Tages-Basis aggregierten Sentiment-Werte und Bitcoin-Renditen.

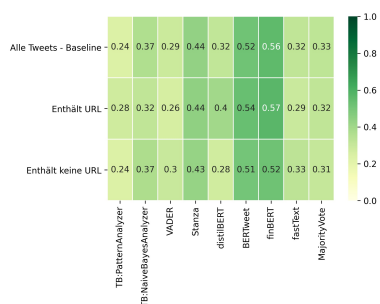


(a) Auf Stundenbasis aggregiert. (b) Auf Tagesbasis aggregiert.

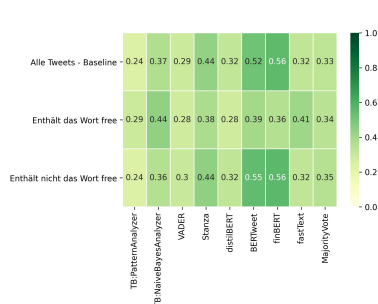
Abbildung A.2.: Pearsonsche Korrelationskoeffizienten der auf Stunden- und Tages-Basis aggregierten Sentiment-Werte und Bitcoin-Renditen.

A.3. Experiment Teil 2

A.3.1. Tweet-basierte Merkmale



(a) Enthält URL



(b) Enthält „free“



(c) Art des Tweets

Abbildung A.3.: Distanzkorrelationen der durch Tweet-basierte Merkmale resultierenden Teilmengen.

A.3.2. Autoren-basierte Merkmale

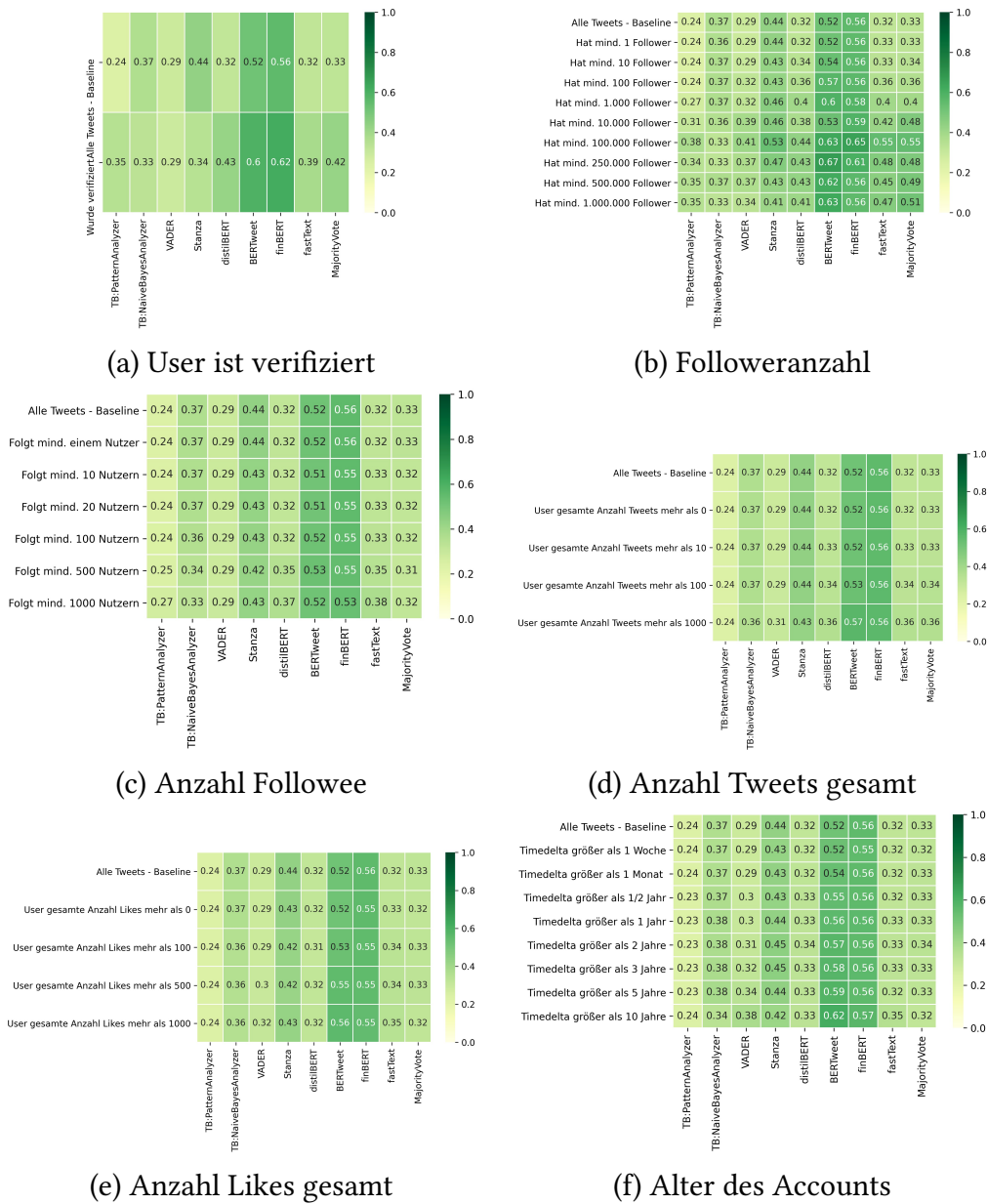


Abbildung A.4.: Distanzkorrelationen Autoren-basierte Merkmale.

A.4. Experiment Teil 3

Tabelle A.2.: Distanzkorrelationen der auf Tagesbasis aggregierten und durch Kombination der Merkmale gefilterten Sentiment-Werte und der Bitcoin-Renditen.

| Sentiment-Klassifizierer | Merkmal 1 | Merkmal 2 | Distanzkorrelation |
|--------------------------|------------------------------|---------------------------------|--------------------|
| TB:PatternAnalyzer | Hat \geq 1 Mio. Follower | Folgt \geq 1000 Nutzern | 0.548 |
| | Hat \geq 100 Tsd. Follower | Ist normaler Tweet | 0.531 |
| | Hat \geq 500 Tsd. Follower | Folgt \geq 1000 Nutzern | 0.472 |
| TB: NaiveBayesAnalyzer | Enthält URL | $\Delta_{Zeit} > 1$ Jahr | 0.462 |
| | Enthält URL | $\Delta_{Zeit} > 2$ Jahre | 0.458 |
| | Enthält URL | $\Delta_{Zeit} > 3$ Jahre | 0.454 |
| VADER | Hat \geq 1 Mio. Follower | Folgt \geq 1000 Nutzern | 0.503 |
| | Hat \geq 100 Tsd. Follower | Ist normaler Tweet | 0.487 |
| | Hat \geq 100 Tsd. Follower | Wurde verifiziert | 0.442 |
| Stanza | Hat \geq 100 Tsd. Follower | Ist normaler Tweet | 0.592 |
| | Hat \geq 100 Tsd. Follower | $\Delta_{Zeit} > 2$ Jahre | 0.546 |
| | Hat \geq 100 Tsd. Follower | $\Delta_{Zeit} > 1$ Jahr | 0.541 |
| distilBERT | Enthält URL | Wurde verifiziert | 0.579 |
| | Ist normaler Tweet | Hat \geq 100 Tsd. Follower | 0.564 |
| | Enthält URL | Hat \geq 250 Tsd. Follower | 0.535 |
| BERTweet | Ist normaler Tweet | Hat \geq 100 Tsd. Follower | 0.708 |
| | Ist normaler Tweet | Enthält keine URL | 0.706 |
| | Ist normaler Tweet | User ges. Anzahl Likes > 1000 | 0.674 |
| fnBERT | Ist normaler Tweet | Enthält keine URL | 0.718 |
| | Hat \geq 100 Tsd. Follower | User ges. Anzahl Likes > 1000 | 0.679 |
| | Hat \geq 100 Tsd. Follower | User ges. Anzahl Likes > 0 | 0.657 |
| fastText | Hat \geq 100 Tsd. Follower | Ist normaler Tweet | 0.611 |
| | Hat \geq 1 Mio. Follower | Folgt \geq 500 Nutzern | 0.609 |
| | Hat \geq 100 Tsd. Follower | $\Delta_{Zeit} > 2$ Jahre | 0.568 |
| MajorityVote | Hat \geq 100 Tsd. Follower | Ist normaler Tweet | 0.607 |
| | Hat \geq 1 Mio. Follower | Folgt \geq 500 Nutzern | 0.581 |
| | $\Delta_{Zeit} > 10$ Jahre | Ist normaler Tweet | 0.572 |

B. Programmdokumentation

In diesem Teil wird kurz aufgeführt durch welche Dateien die in Abb. 5.1 genannten Funktionalitäten abgedeckt werden.

Beschreibungen der jeweiligen möglichen Schritte befinden sich in den betreffenden Jupyter-Notebooks.

Anmerkungen, die für mögliche Weiterentwicklungen hilfreich sein könnten, finden sich in den zugehörigen Python-Skripten.

Voraussetzung für die Nutzung ist das Vorliegen einer lokalen MongoDB-Datenbank, das Vorliegen von Tweets im JSON-Format und die Installation der Python-Pakete in der requirements.txt Datei.

- **Setup:** Importiert die Tweets in die Datenbank, ruft die Sentiment-Klassifikatoren auf und importiert die Klassifikationen, vervollständigt Datensatz um Autorenmerkmale und importiert die Preise in die Datenbank. Betrifft: Setup.ipynb, (api_helper.py, import_helper.py, twint_helper.py, soft_sentiment_helper.py)
- **Experimente:** Führt die in der Arbeit beschriebenen Experimente aus. Betrifft: Experiment1.ipynb, Experiment2.ipynb, Experiment3.ipynb, (mongodb_helper.py, pandas_df_helper.py, analysis_helper.py).
- **Evaluation:** Berechnet die in der Arbeit genannten Evaluationsmaße und führt Tests zur Dauer der Klassifikationen durch. Betrifft: Evaluation.ipynb, (evaluation_helper.py, sentiment_helper.py).

Abbildungsverzeichnis

| | | |
|------|--|----|
| 2.1. | (Beispiel Topologie eines neuronalen Netzwerkes, entnommen aus Kelleher (2019).) | 4 |
| 2.2. | Word2Vec: CBOW-Architektur (links), Skip-gram-Architektur (rechts), entnommen aus Mikolov, Chen u. a. (2013). | 6 |
| 2.3. | Beispiel für Convolutional-Layer, https://www.ibm.com/cloud/learn/convolutional-neural-networks , letzter Zugriff: 26.08.2021. | 7 |
| 2.4. | Beispiel für Max-Pooling, https://cs231n.github.io/convolutional-networks/ , zuletzt aufgerufen am 25.08.2021. | 8 |
| 2.5. | Beispiel für CNN-basierte Sentiment-Klassifikation, entnommen aus Kim (2014). | 9 |
| 5.1. | Grobes Anwendungsfalldiagramm. | 26 |
| 5.2. | Schlusspreise der Handelstage im Zeitraum. | 27 |
| 5.3. | Veränderungen der Schlusspreise zum Vortag. | 28 |
| 5.4. | Durchschnittliche Verteilung der Anzahl von Tweets über den Tagesverlauf. | 28 |
| 5.5. | Verteilung der Tweets über den Zeitraum. | 29 |
| A.1. | Distanzkorrelationen der auf Stunden- und Tages-Basis aggregierten Sentiment-Werte und Bitcoin-Renditen. | 45 |
| A.2. | Pearsonsche Korrelationskoeffizienten der auf Stunden- und Tages-Basis aggregierten Sentiment-Werte und Bitcoin-Renditen. | 45 |
| A.3. | Distanzkorrelationen der durch Tweet-basierte Merkmale resultierenden Teilmengen. | 46 |
| A.4. | Distanzkorrelationen Autoren-basierte Merkmale. | 47 |
| A.5. | Experiment Teil 3: Längen der durch Merkmalskombination induzierten Teilmengen. | 49 |

Tabellenverzeichnis

| | | |
|------|---|----|
| 2.1. | Die fünf größten Kryptowährungen nach Marktkapitalisierung (Quelle: coinmarketcap.com , Stand 19.09.2021) | 11 |
| 4.1. | Wertebereiche Ausgaben der verwendeten Sentiment-Klassifizierer | 15 |
| 4.2. | Tweet-basierte Merkmale & erwartete Auswirkung auf die Distanzkorrelation. | 23 |
| 4.3. | Autor-basierte Merkmale & erwartete Auswirkung auf die Distanzkorrelation. | 24 |
| 5.1. | Ansätze der verschiedenen Sentiment-Analyzer Implementationen. | 30 |
| 5.2. | Vergleich der durchschnittlichen Inferenz-Zeiten der verschiedenen Sentiment-Analyse Lösungen | 34 |
| A.1. | Vergleich Evaluationsmaße der Klassifizierer auf dem erstellten Testset(insg. 500 Tweets, davon 129 negativ; 215 neutral; 156 positiv). | 44 |
| A.2. | Distanzkorrelationen der auf Tagesbasis aggregierten und durch Kombination der Merkmale gefilterten Sentiment-Werte und der Bitcoin-Renditen. . . | 48 |

Literatur

- Ante, L. (2021). „How Elon Musk’s Twitter Activity Moves Cryptocurrency Markets“. In: *Advertising & Marketing Law eJournal*.
- Araci, Dogu (2019). *FinBERT: Financial Sentiment Analysis with Pre-trained Language Models*. arXiv: 1908.10063 [cs.CL].
- Bertschinger, Nils (2020). *PyMLViz*. https://github.com/PyMLVizard/PyMLViz/blob/master/notebooks/GradientDescent_Variants.ipynb. [Online; Zugegriffen am].
- Bishop, Christopher M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag. ISBN: 0387310738.
- Chaudhuri, A. (2006). *Emotion and Reason in Consumer Behavior*. Elsevier Butterworth-Heinemann. ISBN: 9780750679763. URL: <https://books.google.de/books?id=LNaRAAAIAAJ>.
- Cho, Kyunghyun u. a. (2014). „Learning phrase representations using RNN encoder-decoder for statistical machine translation“. In: *arXiv preprint arXiv:1406.1078*.
- Colianni, Stuart G., S. Rosales und Michael Signorotti (2015). „Algorithmic Trading of Cryptocurrency Based on Twitter Sentiment Analysis“. In:
- Devlin, Jacob, Ming-Wei Chang, Kenton Lee und Kristina Toutanova (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. arXiv: 1810.04805 [cs.CL].
- Duchi, John, Elad Hazan und Yoram Singer (2011). „Adaptive Subgradient Methods for Online Learning and Stochastic Optimization“. In: *Journal of Machine Learning Research* 12.61, S. 2121–2159. URL: <http://jmlr.org/papers/v12/duchi11a.html>.
- Goodfellow, Ian, Yoshua Bengio und Aaron Courville (2016). *Deep Learning*. <http://www.deeplearningbook.org>. MIT Press.
- Hauke, Jan und Tomasz Kossowski (Juni 2011). „Comparison of Values of Pearson’s and Spearman’s Correlation Coefficients on the Same Sets of Data“. In: *Quaestiones Geographicae* 30, S. 87–93. DOI: 10.2478/v10117-011-0021-1.
- Hemati, Wahed und A. Mehler (2019). „LSTMVoter: chemical named entity recognition using a conglomerate of sequence labeling tools“. In: *Journal of Cheminformatics* 11.
- Hochreiter, Sepp und Jürgen Schmidhuber (Dez. 1997). „Long Short-term Memory“. In: *Neural computation* 9, S. 1735–80. DOI: 10.1162/neco.1997.9.8.1735.
- Huang, Yen-Hao u. a. (2019). *EmotionX-IDEA: Emotion BERT – an Affectional Model for Conversation*. arXiv: 1908.06264 [cs.CL].
- Hutto, C.J. und Eric Gilbert (Jan. 2015). „VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text“. In:
- Joulin, Armand, Edouard Grave, Piotr Bojanowski und Tomas Mikolov (2016). *Bag of Tricks for Efficient Text Classification*. arXiv: 1607.01759 [cs.CL].
- Kelleher, J.D. (2019). *Deep Learning*. The MIT Press Essential Knowledge series. MIT Press. ISBN: 9780262354905. URL: <https://books.google.de/books?id=ZU6qDwAAQBAJ>.
- Kim, Yoon (2014). „Convolutional Neural Networks for Sentence Classification“. In: *EMNLP*.

- Kingma, Diederik P. und Jimmy Ba (2017). *Adam: A Method for Stochastic Optimization*. arXiv: 1412.6980 [cs.LG].
- Lhabitant, Francois (Apr. 2011). „Correlation vs. Trends in Portfolio Management: A Common Misinterpretation“. In: *SSRN Electronic Journal* 15. DOI: 10.2139/ssrn.1808267.
- Li, Tianyu Ray, Anup S. Chamrajnagar, Xander R. Fong, Nicholas R. Rizik und Feng Fu (2019). „Sentiment-Based Prediction of Alternative Cryptocurrency Price Fluctuations Using Gradient Boosting Tree Model“. In: *Frontiers in Physics* 7, S. 98. ISSN: 2296-424X. DOI: 10.3389/fphy.2019.00098. URL: <https://www.frontiersin.org/article/10.3389/fphy.2019.00098>.
- Liu, Bing (2015). *Sentiment analysis : mining opinions, sentiments, and emotions*. URL: <https://doi.org/10.1017/CB09781139084789>.
- Liu, Yinhan u. a. (2019). *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. arXiv: 1907.11692 [cs.CL].
- Malo, Pekka, Ankur Sinha, Pyry Takala, Pekka Korhonen und Jyrki Wallenius (2013). *Good Debt or Bad Debt: Detecting Semantic Orientations in Economic Texts*. arXiv: 1307.5336 [cs.CL].
- Manning, Christopher u. a. (Juni 2014). „The Stanford CoreNLP Natural Language Processing Toolkit“. In: *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Baltimore, Maryland: Association for Computational Linguistics, S. 55–60. DOI: 10.3115/v1/P14-5010. URL: <https://aclanthology.org/P14-5010>.
- Metsis, Vangelis, Ion Androutsopoulos und Georgios Paliouras (Jan. 2006). „Spam Filtering with Naive Bayes - Which Naive Bayes?“ In:
- Mikolov, Tomas, Kai Chen, Greg Corrado und Jeffrey Dean (2013). *Efficient Estimation of Word Representations in Vector Space*. arXiv: 1301.3781 [cs.CL].
- Mikolov, Tomas, Wen-tau Yih und Geoffrey Zweig (Juni 2013). „Linguistic Regularities in Continuous Space Word Representations“. In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Atlanta, Georgia: Association for Computational Linguistics, S. 746–751. URL: <https://aclanthology.org/N13-1090>.
- Nakamoto, Satoshi (März 2009). „Bitcoin: A Peer-to-Peer Electronic Cash System“. In: *Cryptography Mailing list at https://metzdowd.com*.
- Ng, Andrew Y. (2004). „Feature Selection, L1 vs. L2 Regularization, and Rotational Invariance“. In: *Proceedings of the Twenty-First International Conference on Machine Learning, ICML '04*. Banff, Alberta, Canada: Association for Computing Machinery, S. 78. ISBN: 1581138385. DOI: 10.1145/1015330.1015435. URL: <https://doi.org/10.1145/1015330.1015435>.
- Nguyen, Dat Quoc, Thanh Vu und Anh Tuan Nguyen (2020). „BERTweet: A pre-trained language model for English Tweets“. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, S. 9–14.
- Qi, Peng, Yuhao Zhang, Yuhui Zhang, Jason Bolton und Christopher D. Manning (2020). „Stanza: A Python Natural Language Processing Toolkit for Many Human Languages“. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. URL: <https://nlp.stanford.edu/pubs/qi2020stanza.pdf>.

- Rao, T. und S. Srivastava (2012). „Analyzing Stock Market Movements Using Twitter Sentiment Analysis“. In: *ASONAM 2012*.
- Rosenthal, Sara, Noura Farra und Preslav Nakov (Aug. 2017). „SemEval-2017 Task 4: Sentiment Analysis in Twitter“. In: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Vancouver, Canada: Association for Computational Linguistics, S. 502–518. DOI: 10.18653/v1/S17-2088. URL: <https://aclanthology.org/S17-2088>.
- Sanh, Victor, Lysandre Debut, Julien Chaumond und Thomas Wolf (2020). *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*. arXiv: 1910.01108 [cs.CL].
- Sebastiani, Fabrizio (März 2002). „Machine Learning in Automated Text Categorization“. In: *ACM Comput. Surv.* 34.1, S. 1–47. ISSN: 0360-0300. DOI: 10.1145/505282.505283. URL: <https://doi.org/10.1145/505282.505283>.
- Smedt, Tom De und Walter Daelemans (2012). „Pattern for Python“. In: *Journal of Machine Learning Research* 13.66, S. 2063–2067. URL: <http://jmlr.org/papers/v13/desmedt12a.html>.
- Socher, Richard u. a. (Jan. 2013). „Recursive deep models for semantic compositionality over a sentiment treebank“. In: *EMNLP* 1631, S. 1631–1642.
- Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever und Ruslan Salakhutdinov (2014). „Dropout: A Simple Way to Prevent Neural Networks from Overfitting“. In: *Journal of Machine Learning Research* 15.56, S. 1929–1958. URL: <http://jmlr.org/papers/v15/srivastava14a.html>.
- Sul, Hong, Alan Dennis und Lingyao Yuan (Juni 2016). „Trading on Twitter: Using Social Media Sentiment to Predict Stock Returns: Trading on Twitter“. In: *Decision Sciences* 48. DOI: 10.1111/dec.12229.
- Székely, Gábor J., Maria L. Rizzo und Nail K. Bakirov (Dez. 2007). „Measuring and testing dependence by correlation of distances“. In: *The Annals of Statistics* 35.6. ISSN: 0090-5364. DOI: 10.1214/009053607000000505. URL: <http://dx.doi.org/10.1214/009053607000000505>.
- Vaswani, Ashish u. a. (2017). „Attention is All You Need“. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems. NIPS’17*. Long Beach, California, USA: Curran Associates Inc., S. 6000–6010. ISBN: 9781510860964.
- Wolf, Thomas u. a. (Okt. 2020). „Transformers: State-of-the-Art Natural Language Processing“. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, S. 38–45. URL: <https://www.aclweb.org/anthology/2020.emnlp-demos.6>.
- Xu, Jiahua, Nazariy Vavryk, Krzysztof Paruch und Simon Coussaert (2021). *SoK: Decentralized Exchanges (DEX) with Automated Market Maker (AMM) protocols*. arXiv: 2103.12732 [q-fin.TR].
- Zhu, Yukun u. a. (2015). *Aligning Books and Movies: Towards Story-like Visual Explanations by Watching Movies and Reading Books*. arXiv: 1506.06724 [cs.CV].