NLP:
Stimmungsanalyse rund um das Thema Ethereum

Betreuer: Tatiana Bladier, Kilian Evang

Verfasst von Aldi Halili: 3013735, Brahim Chouikha: 2837831, Saber Klai: 2846796

1 Inhaltsverzeichnis

2 Einleitung

Sentiment-Analyse, auch Opinion Mining genannt, ist das Studiengebiet, das die Meinungen, Gefühle, Einschätzungen, Einstellungen und Emotionen von Menschen gegenüber Entitäten und ihren in geschriebenem Text ausgedrückten Attributen analysiert. Die Entitäten können Produkte, Dienstleistungen, Organisationen, Einzelpersonen, Ereignisse, Probleme oder Themen sein (Lui, 2015).

Dieses Projekt beschäftigt sich mit dem zentralen Thema des „Natural Language Processing". Das Projekt behandelt diesen Bereich der NLP, indem es sich näher mit dem Aspekt „Opinion Mining" befasst. Das bedeutet, es wird untersucht, wie es mit der Hilfe eines Algorithmus ermöglicht werden kann, einen Textkorpus zu bearbeiten und den emotionalen Ton hinter dem Textkorpus zu identifizieren. Man möchte eine augenblickliche allgemeine Meinung der Textverfasser ermitteln können.

Der Hintergrund für dieses Projektes war die Überlegung, ob es möglich sein könnte herauszufinden, wie sich im Allgemeinen die Meinungen auf dem aktuellen Krypto Markt bezüglich der Kryptowährung Ethereum verhalten, um daran angelehnt eine Investitionsentscheidung zu treffen. Die Hypothese ist hierbei, dass es möglich sein sollte, eine allgemeine positive, negative oder neutrale Haltung zu der Thematik „Ethereum" aus Texten herauszuarbeiten. Fraglich ist hierbei, wie nachvollziehbar und akkurat diese Klassifizierung einer Emotion durch das Verfahren sein kann.

Das Projekt wird zur Überprüfung der Hypothese einen Textkorpus verwenden, der aus 1084 Tweets besteht. Die Tweets werden durch „Scraping" von Twitter Nutzeraccounts, die sich auf der Plattform mit dem Thema Kryptowährung beschäftigen, heruntergeladen. Dieser Korpus wird mittels der NLP-Tools spacy und Regex bereinigt und vorbereitet. Im Anschluss wird dieser bereinigte Datensatz mittels „VADER", ein Algorithmus zur Stimmungsanalyse, weiterbearbeitet. Zusätzlich wird das Modul „Wordcloud" verwendet. Diese beiden Schritte sollen die ermittelten Tendenzen des Textkorpus zusätzlich veranschaulichen.

Das Projekt erwartet eine eindeutige Ermittlung der aktuell herrschenden Tendenzen in der Kommunikation der Interessenten an Kryptowährung wie Ethereum.

3 Methoden

IN diesem Projekt wurden verschiedenen Module von Python verwendet, die die Bearbeitung des Textkorpus und die Ermittlung der ausgedrückten Empfindungen ermöglichen. Die Module sind „Tweepy“, „Spacy“, „VADER“ und „Wordcloud.“

3.1 Tweepy:

Tweepy[1] ist ein Open-Source-Python-Paket, das es ermöglicht, mit Python auf die Twitter-API zuzugreifen. Die Twitter-API gibt Entwicklern den Zugriff auf die meisten Funktionen von Twitter. Sie können die API verwenden, um Informationen zu Twitter-Entitäten wie Tweets, Benutzer und Trends zu lesen und zu schreiben. API macht folglich HTTP-Endpunkten verfügbar, die sich auf Folgendes beziehen: Tweets, Retweets, Likes, Direktnachrichten, Favoriten, Tendenzen und Medien.

Die Twitter-API verwendet „OAuth“, ein offenes Autorisierungsprotokoll, um alle Anfragen zu authentifizieren. Bevor man die Twitter-API aufrufen kann, müssen Authentifizierungsdaten erstellt und konfiguriert werden. Es müssen also die erforderlichen Authentifizierungsdaten erstellt werden, um die API verwenden zu können. Diese Anmeldeinformationen sind vier Textzeichenfolgen (Kulkarni and Shivananda, 2019):

1 Consumer key
2 Consumer secret
3 Access token
4 Access secret

3.2 SpaCy & Regex:

SpaCy[2] ist eine Open Source Bibliothek für die Programmiersprache Python**,** die für das Natural Language Processing eingesetzt wird. SpaCy ermöglicht den Text automatisch computerbasiert

[1] Die offizielle Webseite von „Tweepy“: https://docs.tweepy.org/en/stable/
[2] Die offizielle Webseite von „spaCy“: https://spacy.io/

analysieren und verstehen zu können. Für die Verarbeitung von Text und das Verstehen des Inhalts bietet spaCy unterschiedliche Funktionen an wie z.B.:

- Tokenisierung: Segmentierung von Text in Wörter und Satzzeichen
- Part-of-speech (POS) Tagging: Zuordnen von Wortarten
- Lemmatisierung: Zuordnung der Grundformen der Wörter
- Dependency Parsing: Zuordnung von Syntaxabhängigkeiten zwischen den identifizierten Token
- Named Entity Recognition (NER): Benennen von Objekten, Personen oder Orten
- Ähnlichkeitsanalyse von Wörtern, Sätzen oder kompletten Texten

Bezüglich unseres Projekts wurde spaCy für drei grundlegende Schritte für die Vorverarbeitung von Tweets angewendet:

3.2.1 Tokenisierung

Die Tokenisierung ist der erste Schritt in jedem NLP-Modell. Die Hauptfunktion ist es, einen Eingabetext automatisch in Einheiten, namens Tokens, zu segmentieren. Ein Token kann ein Wort, eine Zahl oder ein Satzzeichen sein.

Für die Tokenisierung in diesem Projekt wurde spaCy Bibliothek durch Laden eines englischen Models von spaCy Objekt verwendet. spaCy teilt den Text automatisch in Token auf, wenn ein Text mit dem Modell erstellt wird. Ein Token bezieht sich auf einen einzelnen Teil eines Satzes, der einen semantischen Wert hat.

Durch Anwendung dieses ersten Schrittes (Tokenisierung) wurde festgestellt, dass die Wörter wie *aren't* und *don't* getrennt, betrachtet werden. Das bedeutet, dass *are* und *n't* bzw. *do,* und *n't* als zwei unterschiedliche Wörter auftauchen. Das könnte bei Sentiment Analyse ein Nachteil sein, weil Tokens wie *do, n't, too,* und *much* usw. durch den nächsten Schritt der Stoppwörter-Entfernung entfernt werden und somit die Bedeutung des Satzes beeinflussen (siehe Tab. 1).

In der Fig. 1. wird die Tokenisierung von unserem Korpus dargestellt. Nach der Tokenisierung wurde der Text mithilfe der „join()“ Methode in einen String

umgewandelt und wird als Eingabe für den nächsten Vorverarbeitungsschritt der Stoppwörter-Entfernung angewendet.

3.2.2 Stoppwörter Entfernung

Im englischen Wortschatz gibt es viele Wörter wie *I*, *the*, *are*, und *you*, die sehr häufig im Text vorkommen, aber sie tragen keine wertvollen Bedeutungen für NLP. Es wird empfohlen die Stoppwörter im Rahmen der Verarbeitungsschritte zu entfernen. Durch Entfernung der Stoppwörter verringert sich die Größe der Textkorpus und erhöht sich somit die Leistung des NLP-Modells. Dies könnte für unser Projekt jedoch auch nachteilig sein, weil durch Stoppwörter Entfernung die Bedeutung des Textes (Tweets) geändert werden kann. Wenn man zum Beispiel den Tweet 742 " *she's cute but doesn't trade cardano and…* " betrachtet, stellt man nach der Entfernung der Stoppwörter fest, dass der Satz, der zunächst als negativ bewertet wurde, nach der Entfernung der Stoppwörter positiv bewertet wird.

In unserem Projekt haben wir spaCy Bibliothek importiert. Wir laden das englische Sprachmodell des spaCy-Objekts und speichern die Liste der Stoppwörter in einer Variable. Anschließend erstellen wir eine leere Liste, um Wörter zu speichern, die keine Stoppwörter sind. Mit einer for-Schleife, die über den Text, nach der Tokenisierung iteriert, überprüfen wir, ob das Wort in der Stoppwörterliste vorhanden ist, wenn nicht, fügen wir es der Liste hinzu. Schließlich bekommen wir die Liste der Wörter, die keine Stoppwörter enthalten, indem wir die Funktion "join()" verwenden, und somit haben wir eine endgültige Ausgabe, bei der alle Stoppwörter aus der Zeichenfolge entfernt werden. Diese Ausgabe wird als Eingabe für den letzten Vorverarbeitungsschritt (Lemmatisierung) angewendet.

3.2.3 Lemmatisierung

Lemmatisierung ist der letzte Vorverarbeitungsschritt in diesem Projekt, in dem ein Wort in sein Lemma repräsentiert wird. Ein Lemma ist normalerweise die Wörterbuchversion eines Wortes z.B.: die Wörter *schreiben*, *geschrieben* und *schreibt* haben alle das gleiche Lemma des Wortes *schreiben*.

In das geladene englische Sprachmodel geben wir den Text (Tweets) ein, der in einer Variable gespeichert wird. Mit einer for-Schleife iterieren wir über den Text und mit der ‚.lemma_' Funktion bekommen wir das Lemma für jedes Wort. Die Lemmatisierungs-Ausgabe wird als Eingabe für den VADER-Algorithmus verwendet.

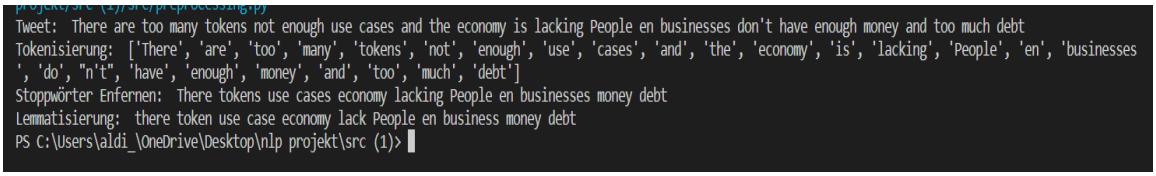

Fig. 1 Hier wird dargestellt, wie ein Tweet sich durch alle Verarbeitungsschritte verringert.

Regex wird in dem Projekt angewendet, um die Tweets zu reinigen. Die Tweets kommen aus sozialen Medien, wobei auf die Schreibweise nicht beachtet wird, und auch viele Sonderzeichen im Text vorkommen. Daher spielt die Entfernung der Sonderzeichen bei der Datenvorbereitung eine wichtige Rolle. Gleichzeitig bringt die Extrahierung der Daten aus dem Internet eine große Herausforderung bezüglich der Datenreinigung mit sich. Was unsere Tweets angeht, besteht der Bedarf an Entfernung von vielen Sonderzeichen wie zum Beispiel: @mentions, hyperlinks, websiten links wie zum Beispiel http/https/www. Andere Sonderzeichen die durch Regex entfernt worden sind, sind: #&\()*+,-/:;@[\]^_`{|}~ξ. Unsere Tweets enthalten auch Emoticons, die durch Regex entfernt wurden.

Ausrufzeichen, Fragezeichen und Emoticons können die Bedeutung eines Satzes hinsichtlich des VADER-Algorithmus verändern. Aus Diesem Grund speichern wir die vorbereiteten Daten in zwei unterschiedliche Dateien. In einer sogenannten „preprocessed_minimal" Datei werden sowohl Ausrufzeichen, Fragezeichen, Dollarzeichen, als auch Emoticons beibehalten. Hier werden auch die mit spaCy möglichen Vorverarbeitungsschritte nicht angewendet. In einer zweiten „preprocessed_maximal" Datei wird der Vorverarbeitungsschritt in jeder Hinsicht vollständig durchgeführt. Auch die Groß- und Kleinschreibenwird in dieser Vorverarbeitung berücksichtigt. Die Funktion „remove_double_tweets" sorgt bei beiden Dateien dafür, dass die Duplikate entfernt werden.

4 *VADER*:

Valence Aware Dictionary and sEntiment Reasoner oder „VADER“[3] ist ein regelbasiertes Modell zur Stimmungsanalyse. VADER fungiert als Lexikon. Es kann Vokabeln, Abkürzungen, Großschreibungen, wiederholte Satzzeichen, Emoticons usw., die normalerweise auf Social-Media-Plattformen verwendet werden, um die eigene Stimmung auszudrücken, effizient verarbeiten, was es zu einer Lösung für die Textanalyse der Stimmung in den sozialen Medien macht (Kulkarni and Shivananda, 2019).

VADER hat den Vorteil, die Stimmung eines beliebigen Textes zu bewerten, ohne dass eine vorherige Schulung, wie es bei Machine-Learning-Modellen der Fall sein kann, erforderlich ist. Das von VADER generierte Ergebnis ist ein Wörterbuch mit 4 Schlüsseln: „neg“, „neu“, „pos“ und „compound“. Die Bezeichnungen „neg“, „neu“ und „pos“ bedeuten jeweils negativ, neutral und positiv. Ihre Summe sollte bei Float-Operation gleich 1 oder nahe daran sein. Die Bezeichnung „compound“ entspricht der Summe der Wertigkeiten jedes Wortes im Lexikon und bestimmt den Grad der Stimmung und nicht den tatsächlichen Wert im Gegensatz zu den vorherigen. Sein Wert liegt zwischen -1 (extremste negative Stimmung) und +1 (extremste positive Stimmung). Die Verwendung der zusammengesetzten Punktzahl kann ausreichen, um die zugrunde liegende Stimmung eines Textes zu bestimmen, denn für:

- eine positive Stimmung, Compound $\geq 0{,}05$
- eine negative Stimmung, Verbindung $\leq -0{,}05$
- eine neutrale Stimmung, die Verbindung liegt zwischen $]-0{,}05, 0{,}05[$

1.1 Wordcloud:

Wordcloud[4] ist im Grunde eine Datenvisualisierungstechnik zur Darstellung von Textdaten. Es geht darum die Häufigkeit von Wörtern in einem Text darzustellen, wobei die Größe jedes Wortes seine Häufigkeit oder Bedeutung angibt. Signifikante Textdatenpunkte können mit einer Wortwolke hervorgehoben werden. Wortwolken werden häufig zur Analyse von Daten von Websites sozialer Netzwerke verwendet (Kulkarni and Shivananda, 2019).

[3] Eine ausführliche Beschreibung von VADER: https://github.com/cjhutto/vaderSentiment/blob/master/README.rst
[4] Die offizielle Webseite von „wordcloud“: https://pypi.org/project/wordcloud/

Jedes Wort in der Wolke hat eine variable Schriftgröße und einen variablen Farbton. Somit hilft diese Darstellung dabei, hervorstechende Wörter zu bestimmen. Eine größere Schriftgröße eines Wortes zeigt seine Hervorhebung in Relation zu anderen Wörtern im Cluster. Die Anzahl der Wörter spielt beim Erstellen einer Wortwolke eine wichtige Rolle. Eine größere Anzahl von Wörtern bedeutet nicht immer eine bessere Wortwolke, da sie unübersichtlich und schwer zu lesen ist. Eine Wortwolke muss immer semantisch aussagekräftig sein und darstellen, wofür sie gedacht ist.

5 Ergebnisse:

Das Ergebnis des durchgeführten Projektes lässt sich anhand der Darstellungen Fig 2., Fig. 3., Fig. 4., und Fig.5 erläutern.

In Fig. 2 wird das sogenannte „preprocessed_minimal“ Datei für den VADER-Algorithmus angewendet, wobei außer mentions, hyperlinks und websiten links nichts entfernt worden ist und Vorverarbeitungsschritt nicht übernommen worden sind.
Es wurden 1084 Tweets als Textkorpus durch VADER analysiert. Das Ergebnis der Analyse sind 436 Zählungen, für die durch VADER identifizierte Empfindung „neu“, 502 Zählungen für die durch VADER identifizierte Empfindung „pos“ und 146 Zählungen für die durch VADER identifizierte Empfindung „neg“ hinsichtlich „Ethereum“.

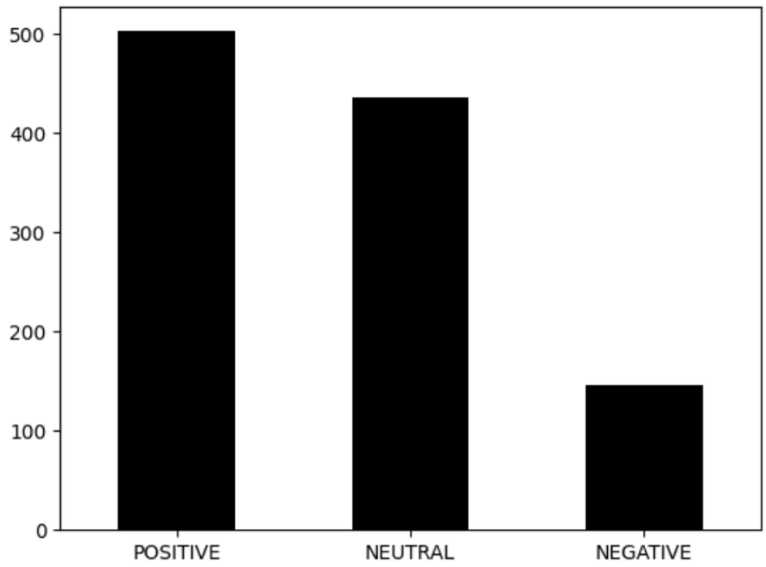

Fig. 2 minimal preprocessed Datei (Zälungen)

In Fig.3 ist das Ergebnis abgebildet für den Durchlauf, bei dem alle Vorverarbeitungsschritte des Korpus übernommen worden sind und der Text vollständig bereinigt wurde. Generiert wurden folgende Ergebnisse: 460 Zählungen „neu“, 486 Zählungen „pos“ und 138 Zählungen „neg“.

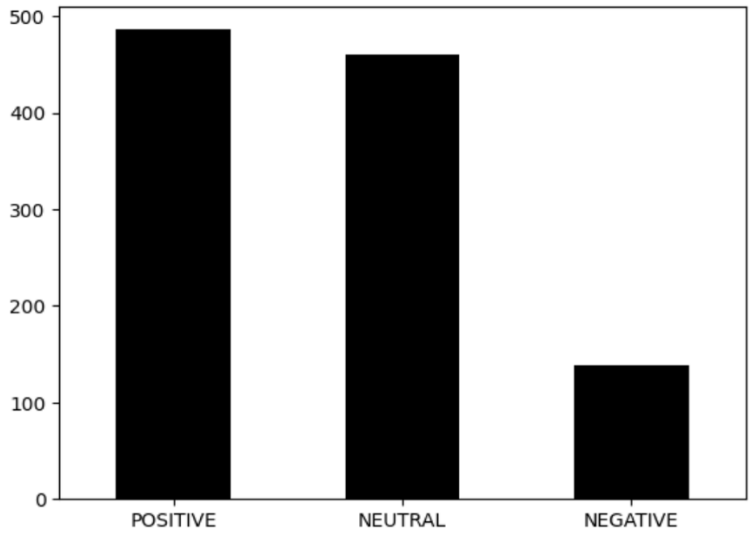

Fig. 3 maximal preprocessed Datei (Zählungen)

In Fig 4. ist erkennbar, dass auf den durch VADER verarbeiteten Textkorpus aus allen Tweets von der „preprocessed_minimal“ Datei über „Ethereum“, 41% „neu“, 46% „pos“ und 13% „neg“ Empfindungen entfallen.

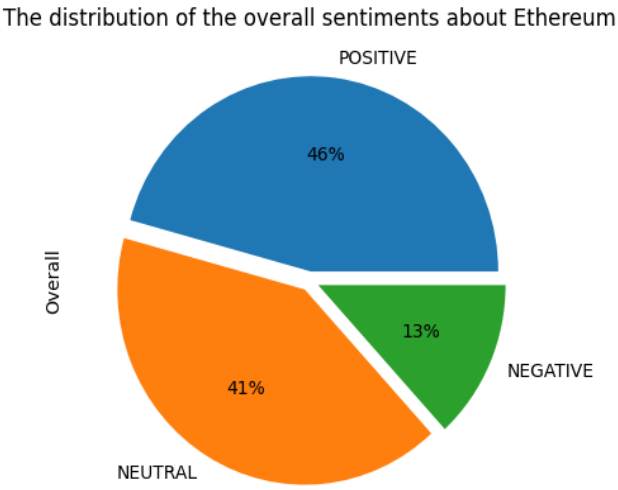

Fig. 4 „preprocessed_minimal“ Datei

Für den durch VADER verarbeiteten Textkorpus aus allen Tweets von der „preprocessed_maximal“ Datei über „Ethereum“ liegen folgende Ergebnisse vor: 43% „neu“, 44% „pos“ und 13% „neg“. Siehe Fig. 5.

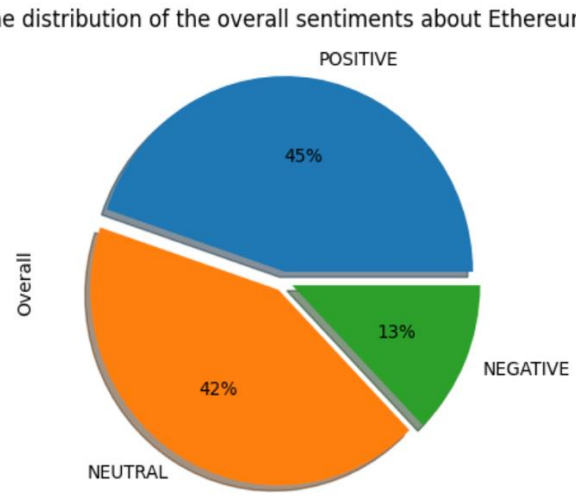

Fig. 5 „preprocessed_maximal“ Datei

Die Wordcloud (Fig. 6) gibt den folgenden Worten eine Hervorhebung: „ethereum“, „nft“, „eth“, „nftcommunity“, „will“, „crypto“, „bitcoin“, „price“, „bought“ und „sold“.

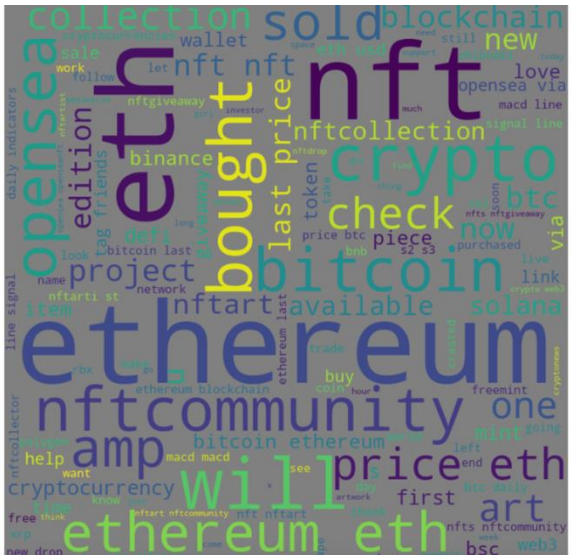

Fig. 6 Wordcloud

6 Diskussion:

Die Resultate des Projektes zeigen auf den ersten Blick die erhoffte erzielte Verteilung von Emotionen innerhalb der Textkorpora (minimal und maximal) zum Thema Ethereum: 41% - 42% „neu“, 46% - 45% „pos“ und 13% - 13% „neg“. Das Ergebnis in Prozentzahlen ließe, unhinterfragt, somit schlussfolgern, dass die Krypto Community einer Investition in Ethereum allgemein befürworten würde.

Allerdings würden Kenner des aktuellen Marktes und des Krypto Marktes vermutlich zu einer genteiligen Prognose gelangen. Ein Sachkundiger würde wesentlich umfangreichere Aspekte in seine Bewertung einfließen lassen, wie z.B. die aktuelle weltpolitische und wirtschaftliche Lage.

Es gilt wohl auch bei der Diskussion um die Resultate zu bedenken, dass der Einsatz eines Algorithmus wie VADER oder eines Modules wie Wordcloud dazu dienen kann einen ersten Eindruck zu vermitteln, welcher jedoch folgend von einem Leser eingeordnet, validiert oder falsifiziert werden muss. Dies kann nicht über den Einsatz eines einzigen „tools“ erfolgreich gelingen. Eine Kombination aus mehreren zusätzlichen Ansätzen und Aspekten (Börsenkurse, Preisbewegungen, politische und gesellschaftliche Faktoren wie der Krieg in der Ukraine oder ein Tweet von Elon Musk) erscheint notwendig um die Ergebnisse der Analyse/n einordnen zu können. Außerdem wird diese Einordnung, Validierung oder Falsifizierung immer im Kontext der herangezogenen Zusatzinformationen des Lesers stehen und ist daher dann auch ein individuelles Ergebnis.

Feststellen kann man jedoch, dass die Benutzung von Wordcloud einem nicht Sachkundigen, aber auch einem Kenner, zum Thema Ethereum einen guten ersten Eindruck geben kann. Das Modul liefert passend zum Korpus eine Reflektion der Hauptthemen, der Kernbegriffe und der Grundstimmung zu dem besagten Thema. Das Ergebnis ist für einen Leser intuitiv erfassbar, was jedoch den Betrachter auch auf einer unterbewussten und emotionalen Ebene anspricht. Auch hier ist daher zu bemerken, dass eine Betrachtung des Ergebnisses von Wordcloud ebenfalls eine individuelle Interpretation der gelieferten Informationen ist. In diesem Kontext kann das Modul daher lediglich dazu dienen einen Überblick über den Korpus und eine Tendenz der Korpusdynamik zu vermitteln. Das ist grundsätzlich nichts Schlechtes, jedoch für die Fragestellung des Projektes nicht weiter hilfreich, da man eine Investitionsentscheidung auf

dieser Basis wohl kaum treffen würde, es sei den man wolle tatsächlich dem ersten Eindruck und der persönlichen Wahrnehmung trauen.

In der Projektarbeit stellte sich der Korpus und die Qualität des Korpus als eine Herausforderung dar. Dabei fallen zwei wesentliche Kritikpunkte ins Gewicht:
Erstens muss erwähnt sein, dass bei näherer Betrachtung des Korpus deutlich wurde, dass eine drastische Mehrheit der Tweets auch nach Standards des Natural Language sehr zweideutig oder mehrdeutig sind. Die Community scheint dazu zu tendieren nicht nur sehr verkürzt Sprachelemente einzusetzen, sondern auch eigene interne Jargons und geprägte feststehende Redewendungen zu verwenden. VADER scheitert nach Durchsicht an der Mehrheit der Tweets. Das verfälscht drastisch VADERs Gesamtergebnis.
Zweitens kann VADER nach Durchsicht der Ergebnisse, in vielen Fällen nicht die Korrekte Zuordnung der festgestellten Emotion leisten. Im folgenden Beispiel wird dies deutlich. VADER hat hier, korrekterweise, eine positive Emotion ermittelt. Diese ist jedoch bei genauerer Durchsicht nicht auf Ethereum bezogen, sondern auf Binance Smart Chain. Dieser Tweet geht daher fälschlicherweise mit der Bewertung „pos“ in die 46% der overall distribution of sentiments about ethereum ein. VADER hat also die Kategorie „pos“ an dieser Stelle korrekt ermittelt, jedoch eine das Ergebnis verfälschende Zuordnung vorgenommen.

Beispiel:
More and more DApps are starting to build their products on the Binance Smart Chain because the block time of Binance Smart Chain is as fast as 5 seconds and the gas fees are much cheaper compared to Ethereum ETH

Eine weitere Beobachtung bezieht sich auf den Korpus und das preprocessing. Im Projekt wurde versucht durch preprocessing des Korpus eine Verbesserung der Versuchsergebnisse zu erzielen. Bei genauerer Durchsicht der Ergebnisse von VADER ergab sich jedoch folgendes Bild. VADER scheint den minimal preprocessed Korpus korrekter verarbeiten zu können als den maximal preprocessed Korpus. Dieses Ergebnis war nicht erwartet worden. Es war angenommen worden, dass preprocessing für die Anwendung von VADER zielführend sein würde. Dies hat sich nicht bewahrheitet. Es wird vermutet, dass preprocessing für VADER wichtige Informationen entfernte (bereinigte) die jedoch bei der Ermittlung von Emotionen dienlich sind, wie z.B. Emoticons, Satzzeichen und Sentimnentwords. Da VADER mit einem regelbasierenden Wörterbuch mit Gewichtung arbeitet, um Emotionen aufzuzeigen, sind

gerade diese Informationen, die maximal preprocessing bereinigt, für VADER Hauptinformationsgeber. Dies verdeutlichte sich bei näherer Durchsicht der Ergebnisse und des Korpus wie auch die Beispiele demonstrieren.

Mininal preprocessed:

"*she's cute but doesn't trade cardano and lost money on ethereum making her a 2 in my book nftcommunity nftcannabis polygonpotheads*"

Compound	Positive	Negative	Neutral	Overall
-0.2382	0.084	0.123	0.793	NEGATIVE

Maximal preprocessed:

"*cute trade cardano lose money ethereum make 2 book nftcommunity nftcannabis polygonpothead*

Compound	*Positive*	*Negative*	*Neutral*	*Overall* "
0.0772	0.191	0.172	0.637	POSITIVE

Minimal preprocessed:

"*make a wish 🤌 let your life pass with health first nothing unhealthy happens everything is fine after that just want whatever you want to do and don't stop fighting… 🎈editions 88 🎈edition price 0030 eth*= "

Compound	Positive	Negative	Neutral	Overall
0.8300	0.269	0.000	0.731	POSITIVE

Maximal preprocessed:

"*wish let life pass health unhealthy happen fine want want stop fight edition 88 edition price 0030 eth*"

Compound	Positive	Negative	Neutral	Overall
0.4767	0.270	0.312	0.418	NEGATIVE

7 Zusammenfassung:

Das Projekt startete mit der Überlegung, ob es möglich sein könnte herauszufinden, wie sich im Allgemeinen die Meinungen auf dem aktuellen Krypto Markt bezüglich der Kryptowährung Ethereum verhalten, um daran angelehnt eine Investitionsentscheidung zu treffen.

Die Hypothese war, dass es möglich sein sollte, eine allgemeine positive, negative oder neutrale Haltung zu der Thematik „Ethereum" aus Texten herauszuarbeiten. Das Projekt hat gezeigt,

dass es möglich ist, aus einem Textkorpus eine emotionale Haltung herauszuarbeiten, indem man einen Algorithmus wie VADER anwendet. Es wurde auch ersichtlich, dass bei diesem Prozess einige Herausforderungen entstehen können. Die Wichtigkeit der Beschaffenheit des Korpus wurde hierbei deutlich. Es zeigte sich, dass entgegen der Erwartung, eine Vorverarbeitung des Textkorpus nicht dienlich hierbei war. Es zeigte sich, dass der Algorithmus zwar zuverlässig die Kategorisierung in „pos“, „neg“ und „neu“ vornahm, dass jedoch die Zuordnung zum Textgegenstand der Sentiments nicht immer erfolgreich gelang. Die Darstellung einer prozentualen Verteilung der drei Sentiments ist zwar technisch möglich, jedoch beinhaltet diese nach Kontrolle des Korpus auch Fehler. Die Abweichungen können schwer abgeschätzt werden. Die Darstellung eines emotionalen allgemeinen Eindrucks über ein Modul wie Wordcloud ist möglich, die Lesung des Ergebnisses ist jedoch individuell und spricht den jeweiligen Leser auf einer intuitiven Ebene an.

Festzustellen ist also, dass die Klassifizierung einer Emotion und die Feststellung dieser in einem Textkorpus, wie es das regelbasierte Modul VADER vornimmt möglich ist, dass die Qualität des Ergebnisses jedoch maßgeblich von der Qualität des gefütterten Korpus abhängt und dass eine Fehlerquote zu berücksichtigen ist. Auch ist deutlich geworden, dass Ergebnisse der Module Wordcloud und VADER zwingend in weiteren Kontexten betrachtet werden müssen.

Alles in allem kommt man daher zu dem Schluss, dass eine Investitionsempfehlung durch die Ergebnisse des Projektes nicht gestützt werden kann.

8 Referenzen

Kulkarni, Aksahy and Adarsha Shivananda. *Natural Language Processing Recipes: Unlocking Text Data with Machine Learning and Deep Learning Using Python*. 2019: Springer, New York. doi:10.1007/978-1-4842-4267-4

Liu, Bing. *Sentiment Analysis: Mining Opinions, Sentiment, and Emotions*. 2015: Cambridge Universoty Press, New York.