Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelor-Arbeit

Nicolas Hellwegen

Analyse von Nutzer-Kommentaren der YouTube-Trends

Fakultät Technik und Informatik
Department Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Nicolas Hellwegen

Analyse von Nutzer-Kommentaren der YouTube-Trends

Abschlussarbeit eingereicht im Rahmen Bachelor of Science

im Studiengang Angewandte Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. Olaf Zukunft
Zweitgutachter : Prof. Dr.-Ing. Marina Tropmann-Frick

Abgegeben am 05.11.2020

Nicolas Hellwegen

Thema der Arbeit

Analyse der Nutzer-Kommentare unter YouTube Videos in den Trends

Stichworte

YouTube, Kommentare, Analyse, Sentiment, Sentiment Analyse, Big Data

Kurzzusammenfassung

YouTube bietet eine Trend-Liste mit den aktuell beliebten Videos. In dieser Arbeit werden neben der Analyse des Aufbaus der Liste auch die Kommentare der Nutzer auf ihr Sentiment hin untersucht. Der Sentiment Wert wird mit der Like/Dislike Metrik verglichen und geprüft, ob die Kommentarschreiber ein Video im Vergleich anders bewerten. Ein weiterer Punkt sind die Kriterien, die ein Video für einen Platz auf der Liste erfüllen muss und von YouTube nur vage beschrieben werden.

Nicolas Hellwegen

Title of the paper

Analysis of user comments under YouTube videos in the trends

Keywords

YouTube, Comments, Analysis, Sentiment, Sentiment Analysis, Big Data

Abstract

YouTube offers a trend list with the currently popular videos. In this work, in addition to the analysis of the structure of the list, the comments of the users are also examined for their sentiment. The sentiment value is compared with the like/dislike metric and it is checked whether the comment writers rate a video differently in comparison. A further point are the criteria that a video has to meet to get a place on the list and are only vaguely described by YouTube.

Inhaltsverzeichnis

1 Einleitung

1.1 Motivation

Vor 15 Jahren wurde die Video Plattform YouTube gegründet. Heute ist YouTube eine der meistbesuchten Seiten im Internet. Etwa 2 Milliarden aktive, individuelle Nutzer (Abbildung 1) besuchten die Webseite, um Videos zu schauen, eigene hochzuladen und/oder um Kommentare dazulassen. Die aktuell beliebtesten Videos auf YouTube werden als Trendliste seinen Nutzern angeboten.
In der Kommentar-Sektion unter den Videos wird großzügig kommuniziert. Die Nutzer verwenden dabei das gegebene Videomaterial oder dem Content Creator, welcher das Video hochgeladen hat, als Diskussionsgrundlage, um mit anderen Nutzern zu interagieren. Anders als die anderen Sozialen Netzwerke, haben YouTube Kommentare ein negatives Bild bedingt und werden im Vergleich dazu in der Forschung eher außer Acht gelassen.
Die Kommentare können als eine alternative Form der Bewertung eines Videos genutzt werden, wobei Nutzer ihre Meinung in Form von Kritik oder einfachen Statements kundtun und somit die einfachen „Gefällt mir“ (Likes) oder „Gefällt mir nicht“ (Dislikes) Bewertung ergänzen können. Mit dieser Interaktion ist es möglich Content Creatorn möglich auf sein Publikum einzugehen und deutlicheres Feedback zu seinem Inhalt zu erhalten. Diese erweiterte Form der Bewertung setzt allerdings voraus, dass die Kommentarschreiber Videos gleich oder ähnlich bewerten wie bei der Vergabe von Likes. Kommentare können geschrieben werden, ohne einen Like zu geben und umgekehrt. Es kommt die Frage auf, inwieweit sich die beiden Bewertungsschemata voneinander unterscheiden zu einem Video oder ob sich beide nah genug sind, um sich zu ergänzen und einen Mehrwert zu bilden.
Allzu oft haben zudem die Kommentare jedoch – nach eigener Erfahrung des Autors – keinerlei Bezug zum eigentlichen Inhalt, stellen Eigenwerbung für den eigenen Kanal des Verfassers dar oder sind einfach Spam ohne Mehrwert für den Content Creator. Dabei haben wissenschaftliche Arbeiten gezeigt mit diesen Störfaktoren umzugehen [1]. Doch in welchem Ausmaß decken sich die Erfahrung mit negativen Kommentaren zu den tatsächlichen Gegebenheiten, die mit diesen Verfahren wiedergeben lassen?

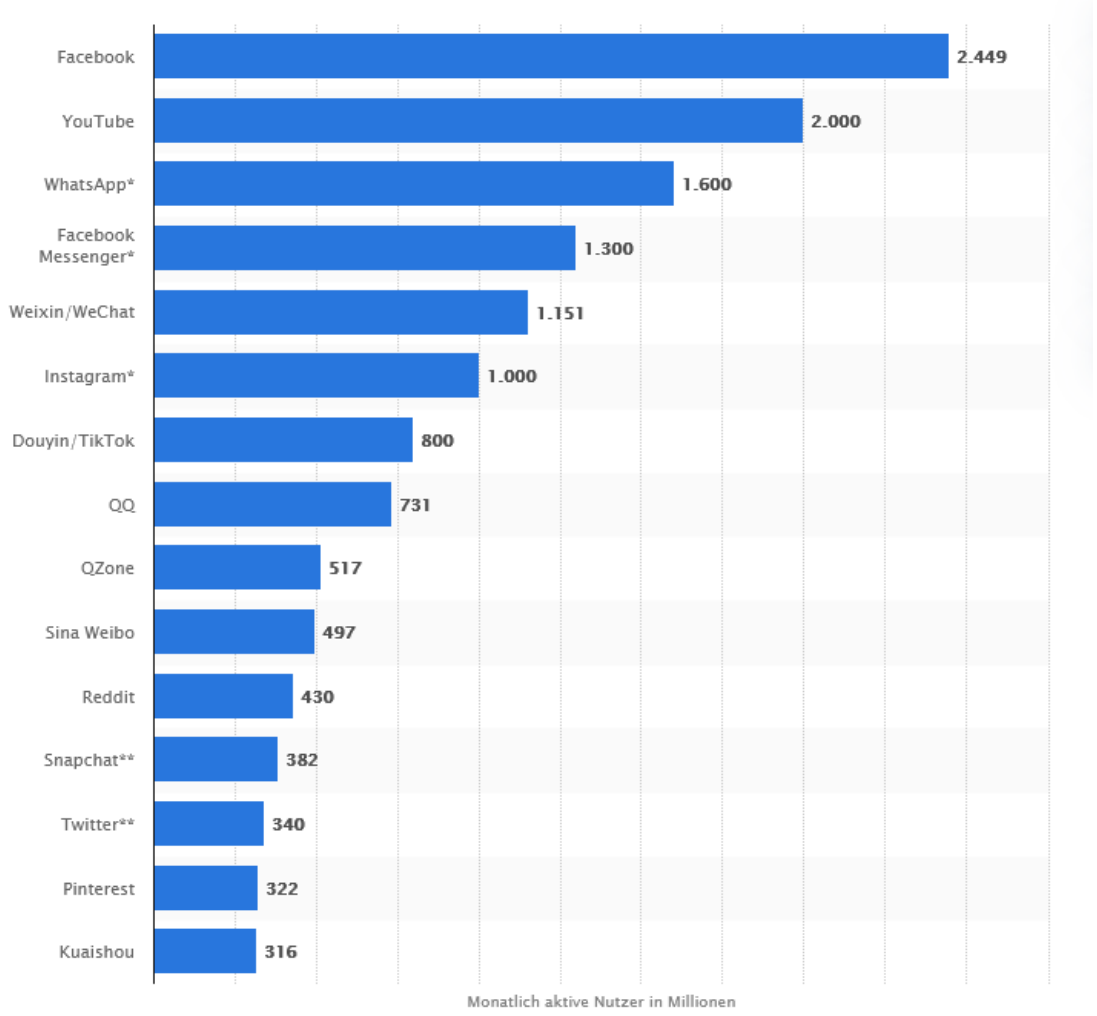

Abbildung 1 Ranking der größten sozialen Netzwerke und Messenger nach der Anzahl der monatlich aktiven Nutzer (MAU) im Januar 2020 [2]

1.2 Ziel der Arbeit

In dieser Arbeit sollen die in der Motivation aufgekommenen Fragen beleuchtet und obendrein geschaut werden, inwieweit man von außen Erkenntnisse mittels einer Software-Lösung über den Trendverlauf gewinnen kann. Als Anhaltspunkt für eine Analyse sollen dabei Metadaten der Videos aber auch die Kommentarsektion dienen. Kommentare sollen auf ihr Sentiment untersucht werden. Weitere Analysen im Bereich Natural Language Processing für die Kommentare sind dabei nicht vorgesehen.

2 Grundlagen

In diesem Kapitel werden die Grundlagen, die dieser Arbeit zugrunde liegen, etabliert und erläutert. Zu den Grundlagen gehören neben der konkreten Erläuterung der Arbeitsbasis noch weitere Arbeiten auf die sich bezogen werden für die Methodiken, die in der Auswertung angewendet werden.

2.1 YouTube-Trends

YouTube kuratiert eine Liste an Videos, die sie als Trends betiteln, welche die aktuell beliebtesten Videos auf der Seite darstellen. Diese sollen, laut YouTube, möglichst „das aktuelle Geschehen in der Welt widerspiegeln“ und „möglichst viele Leute ansprechen“ [3].
Die Liste ist in jedem Land unterschiedlich (mit Ausnahme von Indien, hier wird dieselbe Liste in den 9 gängigsten indischen Sprachen angezeigt) und ändert sich etwa alle 15 Minuten.
In der Regel finden sich Neuveröffentlichung von Künstlern, Filmtrailer, lustige Clips und virale Videos in den Trends.
Im Trend-Tab gibt es neben einer allgemeinen Trend-Liste noch Weitere für unterschiedliche Unterthemen. Über die Browser-Version sind so spezielle Listen explizit für die Kategorien „Musik“, „Gaming“, „Nachrichten“ und „Filme“ zu finden, welche ausschließlich Videos listen, die in der entsprechender Kategorie hochgeladen worden sind. Eine potenzielle Trendplatzierung wird unter dem Titel eines Videos vermerkt, wenn ein Nutzer sich dieses anschaut. Der Vermerk bietet dazu eine Verlinkung zu der Liste, in der das Video platziert worden ist, wobei in der Regel die konkrete Kategorie über den allgemeinen Trends vorgezogen wird.

In der App-Version von YouTube hingegen sind diese Listen unter einem „Entdecken“-Tab zu finden, wobei die allgemeinen Trends mit „Aktuell“ betitelt wurden. Die Umbenennung des „Trend[ing]“-Tabs in „Entdecken“ wurde dabei erst vor kurzem von YouTube vorgenommen und obendrein haben sie angekündigt für weitere Unterthemen Trend-Listen anzubieten. So sollen die Themen „Lernen“ und „Fashion & Beauty“ hinzukommen und die Filme-Kategorie um Fernsehshows erweitert werden [3].

Auf ihrer Hilfeseite nennt YouTube beispielhaft eine Reihe von Kriterien, nach welchen die Videos für diese Liste ausgewählt werden:

1. Anzahl der Aufrufe
2. Wie schnell ein Video Aufrufe generiert
3. Quellen der Aufrufe (auch außerhalb von YouTube)
4. Alter des Videos
5. Leistung des Videos im Vergleich zu anderen Uploads auf demselben Kanal

Des Weiteren sind sie bemüht, möglichst nicht jugendfreie Inhalte (Gewaltdarstellungen, unangemessene Inhalte, Verunglimpfungen etc.) zu filtern. Neben technischen Filtern werden die Videos vorab manuell von Mitarbeitern geprüft, so YouTube weiter auf ihrer Hilfeseite.
Allerdings zeigt sich, dass diese Bemühungen nicht immer erfolgreich sind und es genug Videos in die Liste schaffen, die nicht allzu jugendfrei sind oder anderweitig offensive Inhalte aufweisen.

Insgesamt lässt sich beobachten, dass für YouTube besonders wichtig ist, dass einerseits die Liste möglichst werbefreundlich ist (wie auch jene Videos von Content-Creator, die im YouTube-Partnerprogramm sind) und andererseits Leute mitgezogen werden von der Bewegung, die ein hochplatziertes Video potenziell erzeugt. YouTube zielt darauf ab, dass Nutzer sich ein Video anschauen, dann möglichst lange auf der Seite verbleiben und weitere Videos verfolgen. Ähnlich verhält es sich mit den Videos, die nutzerindividuell auf der Startseite nach Login oder seitlich neben einem aktuell angeschautem Video vorgeschlagen werden.

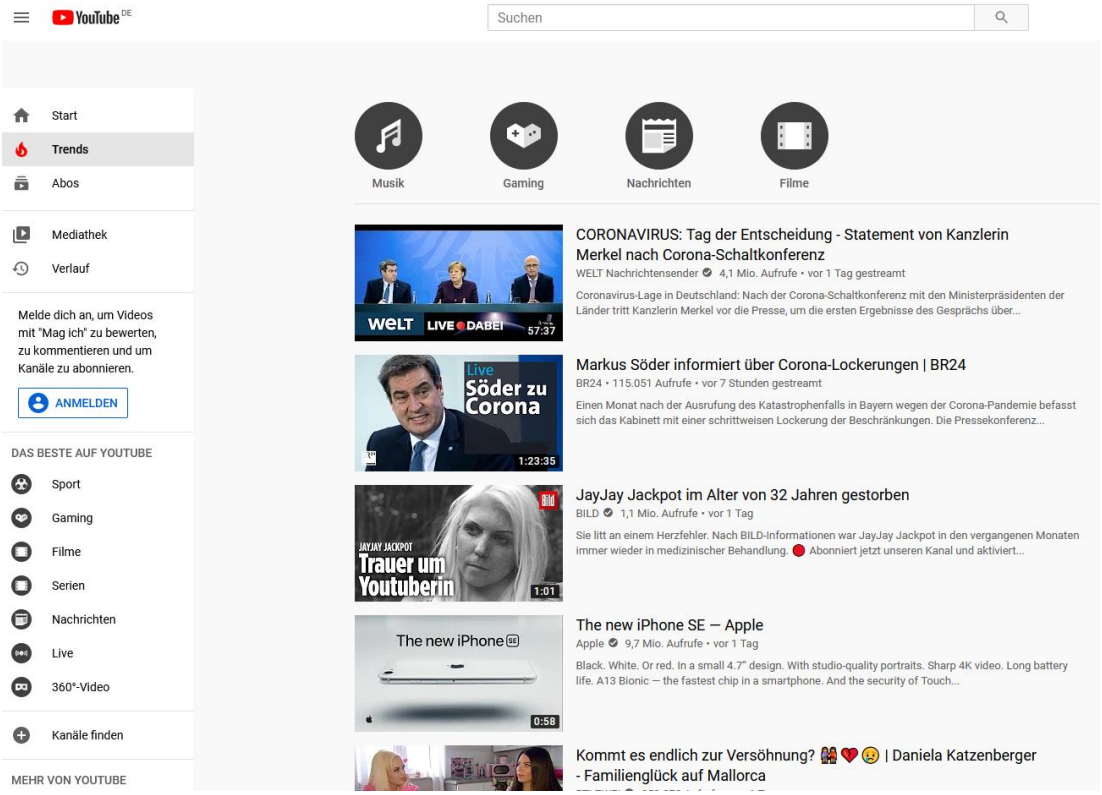

Abbildung 2: Beispiel der deutschen Trends (16.04.2020)

2.2 Kommentar-Klassifizierung

Unter den Kommentaren finden sich einige, die keinerlei Bezug zum eigentlichen Video oder dessen Inhalt haben. Spam, offensive Aufschreie oder Eigenwerbung, um nur ein paar Negativbeispiele zu nennen [1]. Daher wäre eine Methode hilfreich, welche diese Störfaktoren nicht zwangsläufig entfernt, sondern für eine genauere Betrachtung granularer klassifiziert und von jenen, die sich konkret mit dem Video beschäftigen trennt.

2.2.1 Allgemeine Klassifikation

Klassische Methodiken für Klassifikationen stellen in der Regel Verfahren wie Logische Regressionen, k-Nearest Neighbor (kNN) oder Support Vector Machine (SVM) da. Diese Methoden wurden bereits zur Identifikation und Klassifikation von toxischen Kommentaren (Hassrede, Beleidigungen, Drohungen etc.) verwendet und miteinander verglichen [4]. Als Datensatz dienten Wikipedia-Artikel sowie Twitter-Tweets.

2.2.2 YouTube Kommentar Klassifikation

Schultes [1] stellt mit seiner Arbeit dagegen ein anderes Klassifizierungsverfahren vor. Im Gegensatz zu den vorher genannten Verfahren, welche Kommentare anhand ihres Inhaltes klassifizieren, wurden Features definiert, um Kommentare in unterschiedliche Relevanzklassen zu ordnen. Schultes untersuchte wie Nutzer über Kommentare auf YouTube kommunizieren. Hierfür wurden über einen Zeitraum von einem Monat die aktuellen Trendvideos gesammelt und anschließend die Kommentare auf ihre Verteilung innerhalb der von ihm erstellten Kategorisierung untersucht.

Wichtig bei seiner Arbeit ist, dass er sich insbesondere – im Gegensatz zu den anderen genannten Arbeiten – konkret auf YouTube fokussiert. Somit bietet sich Schultes' Klassifikationsverfahren als das am ehesten geeignete Verfahren für diese Arbeit an, da hier keine großen Anpassungen von Nöten sind. Zudem konnte Schultes seine Klassifikation für den Anwendungsfall YouTube erfolgreich verifizieren [1].

Die vorgeschlagenen Relevanzklassen nach Schultes [1]:

- Diskussions-Kommentare (T1): sind Teil einer Diskussion unter Nutzern. Diskussions-Threads entstehen durch die Möglichkeit auf Kommentare anderer zu antworten.

- "Minderwertige" Kommentare (T2): enthalten offensive Aussagen und/oder Beleidigungen, sind somit ohne relevanten Inhalt oder emotionale "Shout-outs".

- "Substantielle" Kommentare (T3): sind Kommentare ohne offensive Aussagen, enthalten gewisse Inhaltsinformationen und sind direkt mit dem eigentlichem Video-Inhalt verbunden.

Schultes nennt die Kategorien T1 und T3 als jene Kommentare, die einen Mehrwert für die Nutzer haben im Gegensatz zu den der T2 Kategorie. In seiner Arbeit konnte er feststellen, dass keine Kategorie groß hervorstach und in den unterschiedlichen Kategorien die Verteilung sich anders gestaltete [1].
Die Features der Kategorien (Tabelle 1) werden ebenfalls aus der Arbeit von Schultes entnommen und ggf. angepasst.
Das Feature „EMOTINAL HINT" wurde von Schultes mit einer selbst erstellten Wortliste erfasst, nachdem Worte mit bestimmten Emotionen beschrieben wurden; „OFFENSIVE HINT" wurde mittels Sentiment Analyse ermittelt, worauf zu einem späterem Zeitpunkt genauer eingegangen wird (siehe 2.3).

Feature	Beschreibung
SPAM HINT	Kommentar wurde als Spam gemeldet
OFFENSIVE HINT	Kommentar hat eine negative Sentiment Stärke größer als 2, ist größtenteils in Großbuchstaben und/oder beinhaltet aggressive Wörter
EMOTINAL HINT	Kommentar beinhaltet „amüsiert“, „begeistert“, „devoted“ oder „angewiderte“ Wörter
EMOTICON HINT	Kommentar beinhaltet mindestens ein Emoticon
PART OF THREAD	Kommentar ist Teil einer Nutzerdiskussion
#WORDS	Anzahl der Worte
TIMESTAMP HINT	Kommentar beinhaltet ein Video Zeitstempel (z.B. „at 1:30“)
KEYWORD MATCH	Kommentar beinhaltet mindestens ein relevantes Schlüsselwort
TITLE MATCH	Kommentar beinhaltet mindestens ein relevante Bestandteil des Video-Titels

Tabelle 1 Feature Beschreibung (nach Schultes) [1]

Klasse		Titel	Definition
T1	C1	Offensive Diskussion Posts	PART OF THREAD & (OFFENSIVE | SPAM)
	C2	Leere Diskussion Posts	PART OF THREAD & (#WORDS < 4)
	C3	Normale Diskussion Posts	PART OF THREAD
T2	C4	Spam oder offensive Posts	SPAM | OFFENSIVE
	C5	Kurze emotionale „Shout-outs“	#WORDS < 9 & (EMOTIONAL | EMOTICON) & !(TIMESTAMP | KEYWORD | TITLE)
	C6	Leere Posts	#WORDS < 4
T3	C7	Referenz zum Video Inhalt	TIMESTAMP
	C8	Beitrag mit Respekt zum Video Inhalt	#WORDS > 8 & (KEYWORD | TITLE)
	C9	Normale Aussage	#WORDS > 10
	C10	Kurze Aussage	#WORDS > 3 & #WORDS < 11

Tabelle 2 Kommentar-Klassen mit formalen Definitionen (nach Schultes) [1]

2.3 Sentiment Analysis

Die Sentiment Analysis (SA) stellt ein klassisches Thema für das Verarbeiten von natürlichen Sprachen dar. Es gilt die etwaige Meinung und/oder Stimmung des Autors eines Textes zu erfassen und für Analysen aufzubereiten.

Allgemein können Sentiment Klassifikationen in Machine Learning-basierte oder Lexikonbasierte Ansätze unterteilt werden. Eine Kombination aus diesen beiden Ansätzen kann ebenfalls in Betracht gezogen werden [5].

Die Textklassifikation über die ML Methode erfolgt über die Reihe an ML Algorithmen, welche SA als ein reguläres Textklassifikationsproblem angeht und sich syntaktische und/oder linguistische Features zunutze macht [6]. In Abbildung 4 finden sich so Lineare Klassifikationen wie SVM wieder, wie auch schon zuvor in 2.2.1 erläutert.

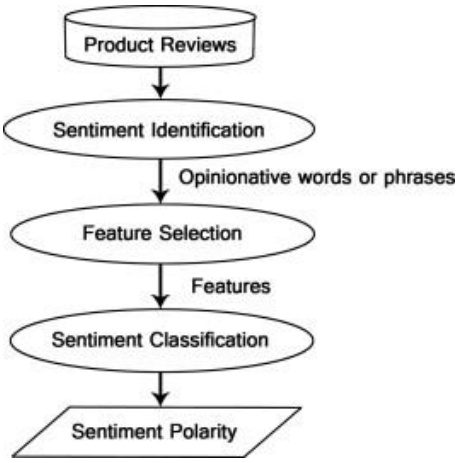

Abbildung 3 Sentiment Analysis Prozess auf Produkt Reviews [11] 5

Der Wörterbuch-basierter Ansatz sammelt hingegen ein kleines Set an „Opinion words“ manuell mit bekannten Sentiment-Werten. „Diese Menge wird dann durch die Suche in den bekannten Korpora WordNet oder Thesaurus nach ihren Synonymen und Antonymen erweitert. Die neu gefundenen Wörter werden einer Seed list hinzugefügt, dann beginnt die nächste Iteration. Der Iterationsprozess stoppt, wenn keine neuen Wörter gefunden werden. Nachdem der Prozess abgeschlossen ist, kann eine manuelle Prüfung durchgeführt werden, um Fehler zu beseitigen oder zu korrigieren“ [6].

Der korpusbasierte Ansatz versucht Meinungsworte mit kontextspezifischen Orientierungen zu finden. Seine Methoden hängen von syntaktischen Mustern oder Mustern, die zusammen mit einer Seed list von „opinion words“ auftreten, ab, um andere „opinion words“ in einem großem Korpus zu finden.

Für beide Varianten werden die Wortlisten manuell erstellt und auf ihr Sentiment bewertet, was einen enormen Aufwand bedeuten kann, wenn man eine hohe Abdeckung (Wörter, die in den untersuchten Anwendungsfall) und Genauigkeit (sind die Sätze entsprechend gewertet worden?) mit seinem Lexikon erreichen möchte.
Für Anwendungsfälle im Bereich Soziale Netzwerke haben einige Lexika Probleme, Sätze mit ihren Sentiments korrekt zu bestimmen, da sie zum großen Teil aus informellen Text wie Akronyme und Emojis/Emoticons bestehen können und sich diese über die Zeit abwandeln können in ihrer Nutzung. Somit müssen die Lexika über die Zeit angepasst und erweitert werden. Alternativ bieten sich hier die kombinierten Ansätze von ML und Lexikon an.

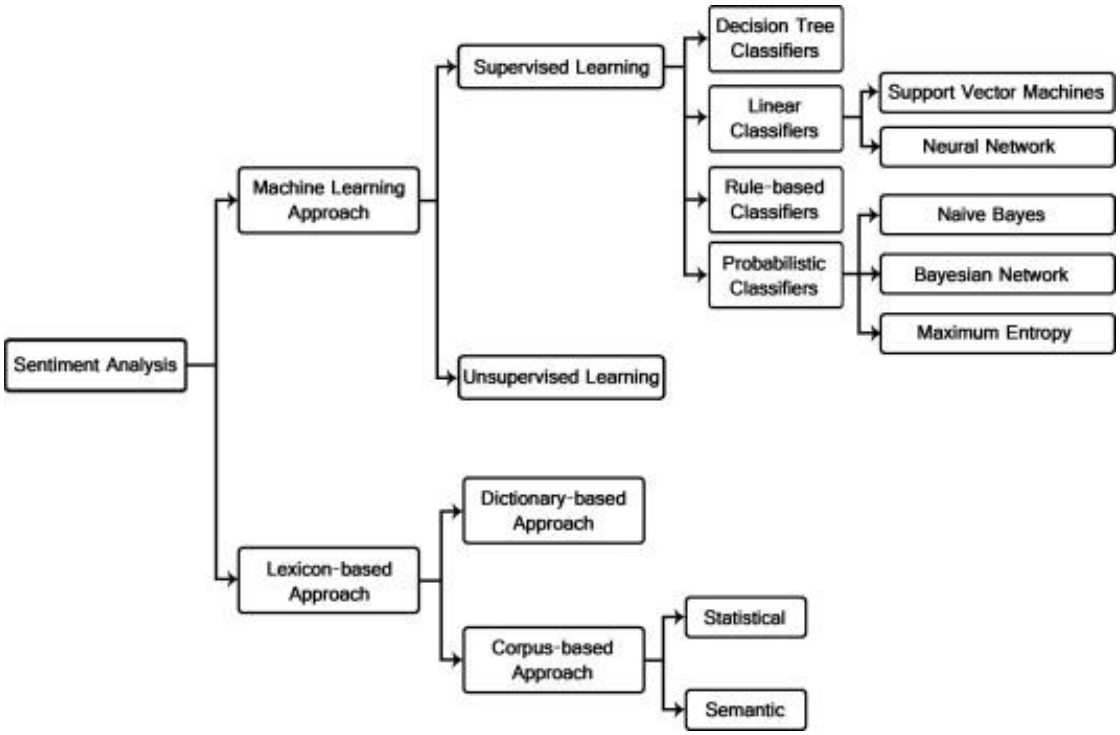

Abbildung 4 Sentiment Analysis Methoden [6]

Ursprünglich hatte Schultes "SentiStrength" von Thelwall für sein Klassifikationsverfahren (siehe 2.2) verwendet. Ebenso kam dieses Programm in Thelwalls eigenen Arbeiten zu den Themen Sentiment Analyse und YouTube zum Einsatz (Quelle Thelwalls Arbeiten) [7, 8].
Seitens Thelwall gab es keine positive Antwort für den Gebrauch von „SentiStrenght" für diese Arbeit, somit musste auf eine Alternative zurückgegriffen werden.

In Betracht wurden zwei Lexika gezogen: SentiWords und VADER

Lexikon 1: SentiWords

SentiWords ist ein Sentiment Lexikon mit rund 155.000 Wörtern der englischen Sprache. Das Lexikon stellt eine Erweiterung von SentiWordNet dar. Die Wörter in diesem Lexikon haben die Form eines Synsets also Sets von lemma#PoS Tupeln und sind mit WordNet-Listen (die Adjektive, Substantive, Verben und Adverbien enthalten) abgeglichen worden [9]. SentiWords zielt auf eine „high coverage" kombiniert mit „good precision" ab [9].
Jedes Wort ist mit einem Positiven und einem negativen Score (0 – 1) assoziiert sowie einem Objektiv oder Neutral Score. Soll das Sentiment für einen Satz ermittelt werden, so werden die korrespondierenden Werte zu den Wörtern im Satz rausgesucht und anschließend der Durchschnitt berechnet.

Lexikon 2: VADER

VADER steht für Valence Aware Dictionary and sEntiment Reasoner und ist ein Lexikon und regelbasiertes SA Tool. Es ist speziell auf den Einsatz für SA von sozialen Netzwerken abgestimmt.
Worte innerhalb des Lexikons haben ein zugewiesenes Sentiment rating (SR) – ein höheres SR deutet auf ein stärkeres Sentiment in die jeweilige Richtung hin. Das Lexikon erzeugt anhand seiner Wortliste vier verschiedene Ratings. Positive, neutral und negative zeigen dabei an, inwieweit der Text in einen der drei Bereiche fällt (prozentual); Compound stellt die standardisierte Summe aller im Text ermittelten SR da und liegt zwischen -1 (absolut negativ) und 1 (absolut positiv). Die 0 steht für neutrales Sentiment.
VADER performt genauso gut oder teilweise sogar besser als ML Ansätze, die speziell für einen Bereich antrainiert wurden [10]. Das Lexikon ist dazu auch noch effizient in seiner Laufzeit. Braucht ein komplexes Modell wie SVM mehrere Stunden (mit Training) oder etliche Minuten (ohne Training), liegt VADER bei einem Bruchteil von Sekunden [10]. Jedoch performt VADER für Twitter-Datensätze besser als für andere Plattformen, bedingt durch den ursprünglichen Fokus der Autoren auf Tweets [11].

Vergleich

Im direkten Vergleich mit anderen Lexika konnte Ribeiro [12] zeigen, dass VADER insgesamt besser abschneidet (Abbildung 5). Getestet wurden die Lexika auf unterschiedlichen Datensätzen wie Tweets, Kommentare (YouTube, NYT, BBC) und Reviews (Amazon, Filme). VADER sticht in diesen Tests insbesondere durch Konsequenz hervor als auf seine hohe Platzierung. Andere Methoden, die im YouTube-Kommentar Datensatz-Test über VADER stehen, sind kostenpflichtige Tools – das bereits erwähnte SentiStrength und LICW15 (Linguistic Inquiry and Word Count Version 2015) – wodurch die Wahl auf VADER fällt.

Anmerkung: 3-classes steht hierbei für Sentiment-Bewertung von negativ, positiv und composite und 2-classes entsprechend nur positiv und negativ. Für ursprüngliche 3-class Lexika wie VADER oder SentiWordNet wurde angenommen, dass neutrale Sätze in einem vorherigem Schritt entfernt wurden (für Vergleiche mit 2-class); für 2-class (LICW oder Emoticons) wurden jene Sätze deren Sentiment nicht ermittelt werden konnte als neutral gekennzeichnet (für Vergleiche mit 3-class) [12].

3-classes			2-classes			
Pos	**Method**	**Mean Rank**	**Pos**	**Method**	**Mean Rank**	**Coverage (%)**
1	VADER	4.00 (4.17)	1	SentiStrength	2.33 (3.00)	29.30 (28.91)
2	LIWC15	4.62	2	Sentiment140	3.44	39.29
3	AFINN	4.69	3	Semantria	4.61	62.34
4	Opinion Lexicon	5.00	4	Opinion Lexicon	6.72	69.50
5	Semantria	5.31	5	LIWC15	7.33	68.28
6	Umigon	5.77	6	SO-CAL	7.61	72.64
7	SO-CAL	7.23	7	AFINN	8.11	73.05
8	Pattern.en	9.92	8	VADER	9.17 (9.79)	82.20 (83.18)
9	Sentiment140	10.92	9	Umigon	9.39	64.11
10	Emolex	11.38	10	PANAS-t	10.17	5.10
11	Opinion Finder	13.08	11	Emoticons	10.39	10.69
12	SentiWordNet	13.38	12	Pattern.en	12.61	65.02
13	Sentiment140_L	13.54	13	SenticNet	13.61	84.00
14	SenticNet	13.62	14	Emolex	14.50	66.12
15	SentiStrength	13.69 (13.71)	15	Opinion Finder	14.72	46.63
16	SASA	14.77	16	USent	14.89	44.00
17	Stanford DM	15.85	17	Sentiment140_L	14.94	93.36
18	USent	15.92	18	NRC Hashtag	17.17	93.52
19	NRC Hashtag	16.31	19	Stanford DM	17.39	87.32
20	LIWC	16.46	20	SentiWordNet	17.50	61.77
21	ANEW_SUB	18.54	21	SASA	18.94	60.12
22	Emoticons	21.00	22	LIWC	19.67	61.82
23	PANAS-t	21.77	23	ANEW_SUB	21.17	94.20
24	Emoticons DS	23.23	24	Emoticons DS	23.61	99.36

Abbildung 5 Durchschnittliche Platzierung verschiedener SA Methoden [12]

3-classes			2-classes			
Pos	**Method**	**Mean Rank**	**Pos**	**Method**	**Mean Rank**	**Coverage (%)**
1	VADER	3.33 (3.60)	1	SentiStrength	1.17 (1.50)	28.29 (24.02)
2	AFINN	4.33	2	Semantria	2.83	61.02
3	Opinion Lexicon	4.33	3	Sentiment140	4.17	36.49
4	Semantria	4.50	4	Opinion Lexicon	6.50	71.59
5	SO-CAL	5.17	5	LIWC15	6.67	65.80
6	LIWC15	6.17	6	AFINN	7.00	74.21
7	Umigon	9.50	7	SO-CAL	7.50	74.59
8	Emolex	10.33	8	VADER	9.50 (9.60)	81.98 (85.34)
9	Sentiment140_L	11.33	9	Umigon	10.50	57.87
10	Stanford DM	11.67	10	Emoticons	11.83	4.99
11	NRC Hashtag	12.00	11	Opinion Finder	13.00	55.66
12	Pattern.en	12.67	12	SenticNet	13.00	95.28
13	SenticNet	13.00	13	USent	14.00	45.66
14	Opinion Finder	13.17	14	NRC Hashtag	14.67	93.43
15	SentiWordNet	13.17	15	Emolex	15.00	69.69
16	SASA	14.67	16	PANAS-t	15.50	5.10
17	SentiStrength	15.17 (19.00)	17	Stanford DM	15.67	84.43
18	Sentiment140	15.50	18	Pattern.en	15.83	59.00
19	USent	15.83	19	Sentiment140_L	15.83	92.30
20	LIWC	17.67	20	SentiWordNet	17.00	63.32
21	ANEW_SUB	17.83	21	SASA	17.50	61.91
22	Emoticons DS	22.67	22	LIWC	19.67	62.24
23	PANAS-t	22.83	23	ANEW_SUB	22.00	94.31
24	Emoticons	23.17	24	Emoticons DS	23.67	99.31

Abbildung 6 Durchschnittliche Platzierung verschiedener SA Methoden für Kommentare [12]

3 Hypothesen

Nachdem im vorherigen Abschnitt die Arbeitsgrundlage etabliert wurde, sollen in diesem Kapitel die Hypothesen für die spätere Analyse aufgestellt werden. Neben den eigentlichen Fragestellungen für die Auswertung wird die Grundlage aufgestellt, wie auf die Hypothesen geschlossen wurde.

3.1 Hypothesenfindung

Die Hypothesen stammen überwiegend aus eigener Überlegung. Sie beschäftigen sich einerseits mit dem, was man aus den Metadaten der Videos (Views, Likes, Kommentarzahl…) und ihrer Platzierung in der Trendliste im Zusammenhang speziell lesen kann.
Andererseits mit den Kommentaren der Nutzer und eine indirekte Bewertung der Videos über diese Form der Interaktion.

3.2 Hypothesen/Auswertung

Für die Auswertung wurden insgesamt 5 Thesen aufgestellt. Sie beschäftigen sich mit der Konstellation der Trends, sowie einem möglicher Einfluss auf Videos; der Zusammensetzung der Kommentare auf Basis der im Grundlagen-Kapitel aufgestellten Kategorisieren und wie Kommentarschreiber indirekt die Videos bewerten anhand von SA.

3.2.1 Höhere Like-Rate gegenüber dem Nutzer-Sentiment

Um die Popularität eines Videos zu messen, werden neben der Like/Dislike Rate, welche anhand ihrer jeweiligen Anzahl unter dem Video zu berechnen ist, noch die Kommentare hinzugezogen und auf ihr Sentiment hin untersucht. Angenommen wird eine positive Like/Dislike Rate für die Trendliste (mehr Likes wie Dislikes), wohingegen die Kommentare in ihrem Sentiment neutral oder potenziell schlechter zu der Like-Rate stehen.

3.2.2 Einblick in die YouTube Trendliste-Kriterien

Im Grundlagen-Kapitel wurde darauf eingegangen nach welchen Kriterien YouTube seine Trendliste gestaltet und Videos auswählt. Diese sind, aus gegebenem Grund, bewusst offener

formuliert, um eine kontrollierte Platzierung seitens der Content-Creator zu vermeiden bzw. zu minimieren.
Nun stellt sich die Frage, ob es möglich ist, mittels der gesammelten Video-Metadaten aus der Trendliste, diese „Kriterien“ nummerisch zu erfassen und einen Blick in die Blackbox zu werfen. Ebenso wird geschaut, ob es Muster in den Videometadaten bezüglich verschiedener Features (wie Laufzeit) zu finden gibt, außerhalb der formulierten Kriterienliste.

3.2.3 Kanäle sind mehrfach mit verschiedenen Videos vertreten

YouTube ermöglicht den Content-Creator Einnahmen über Werbung, welche sie schalten dürfen. Die höchste Anzahl an möglichen Werbe-Clips (innerhalb eines Videos) ist erst ab 10 min Video-Laufzeit möglich. Davor beschränkt es sich auf einen Clip vor oder nach dem Video sowie Banner während des Videos. Eine Möglichkeit, das System für sich zu nutzen, besteht darin eine hohe Upload-Frequenz zu haben. Statt monatlich mindestens täglich ein Video, um den größten Nutzen vom YouTube-Algorithmus zu ziehen und möglichst vielen Leuten empfohlen zu werden. So zumindest gilt es für die Empfehlungen auf der Startseite und der Leiste neben einem aktivgeschautem Video. Somit wird angenommen, dass eine Mehrheit an Kanälen mit mehreren verschiedenen Videos in den Trends zu finden ist als jene die nur ein Video in die Trends bekommen haben.

3.2.4 Kommentare sind gleichverteilt in den Kategorien

Schultes hat in seiner Arbeit zunächst die Hypothese aufgestellt, dass Kommentare bedingt durch einen hohen Anteil negativ Beitragender ein ebenso negatives Bild auf die Nutzer haben. Tatsächlich konnte man in der Arbeit darlegen, dass 30% aller von ihm gesammelten Kommentare aus der T2 Kategorie stammen und diese insbesondere für das negative Bild verantwortlich zu sein scheinen.
Insgesamt stellte sich heraus, dass keine Kommentar-Kategorie in der Gesamtheit überlegen waren. Vielmehr hängt es von der Video-Kategorie und dem Video-Inhalt hab, wie die Nutzer ihre Kommentare darunter verfassen. Diese Ergebnisse sollen versucht werden nachzustellen, wobei die Erwartung für das Endergebnis ähnlich zu Schultes Arbeit liegt.

3.2.5 Höher platzierte Videos haben einen größeren Zuschauerzuwachs

Videos, die auf einem höheren Platz (Top 10) liegen, sollten eher von Nutzern geschaut werden, um den Platz zu entsprechen. Es ist anzunehmen, dass entsprechend platzierte Videos einen höheren Zuschauerzuwachs zu verzeichnen haben. Anders weisen Videos dann in den tieferen Rängen (40 – 50) einen vergleichsweise niedrigen Zuwachs auf.

4 Konzept

Im Konzept-Abschnitt werden einige technische Möglichkeiten für die Anforderungen an die Software sondiert. Die Idee ist eine modular gestaltete Web-Anwendung zu implementieren. Damit soll beispielsweise keine Abhängigkeit für den Nutzer zu einer konkreten Datenbank bestehen.
Analyseauswertungen sollen über die Auswahl der gewünschten Metadaten-Features erfolgen, wobei jede Auswertung ein eigenes Plugin-Modul darstellt.

4.1 YouTube API

Das Sammeln der Videoliste soll über die offizielle YouTube-API erfolgen. In dieser Arbeit wird auf die freie Version verwendet, welche die Zugriffe auf 10.000 Einheiten begrenzt. Dieses Kontingent wird um Mitternacht pazifischer Zeit zurückgesetzt.
Ein Request (Read-only) kostet hierbei jedoch unterschiedliche viele Einheiten, je nach Umfang des Requests. Das Anfragen der Trendliste, welche die 50 beliebtesten Videos inklusive relevanten Metadaten (ID, Statistiken, etc.) beinhaltet, kostet 7 Einheiten.
Resultat des Requests ist eine geordnete Liste (sortiert nach Trendlistenplatzierung). Die Kommentare für ein Video sind ein gesonderter Request (3 Einheiten pro Request), wobei dies sich auf komplette Threads (Kommentar mit eventuellen Antworten auf diese) bezieht als nur auf einen einzelnen Kommentar.
Angedacht ist, dass die Trendliste in einem stündlichen Intervall abgerufen wird. Zwar ändert sich die Liste “etwa alle 15 min” [3] und somit haben sich nach einer Stunde bereits die Positionen 4 mal geändert, allerdings ist das der beste Kompromiss, wenn man noch eine relevante Menge an Kommentare sammeln möchte.

Die YouTube-API gibt bei Anfrage maximal 100 Kommentare in Threads wieder. Die Threads stellen dabei die Antwortmöglichkeit von Nutzern auf einen Kommentar dar. Um mehr als 100 Kommentare von einem Video zu erhalten, wird ein “nextpage” token mitgeliefert, sofern mehr als diese Zahl an Kommentaren vorhanden sind. Dieses ermöglicht bei Weitergabe als Parameter im Request an weitere Kommentare zu kommen.

4.1.1 Alternative Crawling

Alternativ zu den Beschränkungen der YouTube API besteht die Möglichkeit über Crawling an die Daten zu kommen.
Bei dieser Methode entfallen die Zugriffbeschränkungen (wenn gleich man auch bei dieser Methode darauf achten sollte, nicht allzu viel Traffic über eine IP zu generieren), allerdings ist es ein wenig aufwendiger. HTML lässt sich über Pythons Beautiful Soup ziemlich einfach verarbeiten und entsprechende Features herausziehen. Die Kommentare unter den Videos werden allerdings zur Laufzeit per Script generiert, sodass man über virtuelle Browser (beispielsweise Selenium) darauf zugreifen muss.
Dies bedeutet insgesamt einen minimalen Mehraufwand, wenn man mit diesen Tools und dieser Prozedur geübt ist, im Vergleich zu der API.
Allerdings fallen einige Features weg, die nicht auf dem HTML oder via Script geliefert werden, und eventuell von Interesse sein könnten. Es werden etwaige Ländersperren von Videos nur lokal erfasst, während die API eine vollständige Liste liefert, in der ein Video explizit entweder erlaubt oder gesperrt ist. Ist ein Video explizit in einer Reihe von Ländern erlaubt, so kann dieses nur in den genannten Ländern abgerufen werden.
Für Kommentare hingegen wird der Originaltext beim ersten Veröffentlichen über die API zurückgegeben in Kombination mit dem aktuellen, eventuell bearbeiteten Text.

4.2 Datenbank

Angesichts der relativ schnellwachsenden Datenmenge wird eine dokumentenbasierte NoSQL Datenbank verwendet. Dokumentbasiert bietet sich insbesondere an, da die API mit JSONS antwortet und diese einfach abgespeichert werden können ohne weitere Zwischenschritte oder vorab definierte Schemata.
Zur Auswahl stehen hierfür zwei Datenbanken, die in Frage kämen: MongoDB und CouchDB. Eine dritte Variante, Terrastore, wurde ebenfalls in Betracht gezogen, jedoch aufgrund des eingestellten Supports nicht weiter untersucht und ausgelassen [13]. Die gewählten Varianten sind zwar beide dokumentenbasiert, erfüllen aber unterschiedliche Anforderungen, auf die im Folgenden weiter eingegangen wird.

4.2.1 CouchDB

CouchDB ist eine in Erlang geschriebene dokumentenbasierte Datenbank, welche ihre API über HTTP anbietet. CouchDB ist plattformunabhängig und kann unter anderem auch auf Mobilgeräten eingesetzt werden [14].
Die Dokumente werden im JSON-Format und schemafrei gespeichert.
Anbindungen sind Programmiersprachenunabhängig durch die HTTP-API; jede Sprache, die, in der Lage ist einen HTTP-Request zu senden, kann auf die Datenbank zugreifen. Es gibt auch

einige Module (z.B. pycouch für Python) für konkrete Sprachen, welche einige Fälle für den Nutzer abnehmen und das Handling vereinfachen. Replizierung kann bei CouchDB über Master-Master oder Master-Slave betrieben werden und bietet seit neueren Updates Möglichkeiten für Sharding.

4.2.2 MongoDB

MongoDB ist ebenfalls eine schemafreie, dokumentenbasierte Datenbank, welche allerdings in C++ geschrieben wurde [15].
Die Dokumente werden im BSON-Format, eine binäre Form von JSON, gespeichert.
Anbindungen an die Datenbank erfolgen über Treiber für konkrete Programmiersprachen, welcher auf der MongoDB Seite oder entsprechenden Seiten heruntergeladen werden können. MongoDB kann im Master-Slave Modus betrieben werden, um Ausfallsicherheit zu bieten. Master-Master Replizierung unterstützt MongoDB in Form seines Sharding-Features indirekt. Für eine horizontale Skalierung bietet MongoDB das AutoSharding-Feature, mit welchem man den Server auf verschiedenen Maschinen aufteilen kann. Für die Applikation wirkt es weiterhin so, als würde sie mit einer einzigen MongoDB-Instanz kommunizieren.
MongoDB ist die beliebteste dokumentenbasierte DB (Abbildung 7), wodurch eine große, aktive Community gewachsen ist.

4.2.3 Vergleich

Die Wahl der Datenbank gestaltete sich durch die Ähnlichkeit der beiden Optionen als schwierig da. Letztlich war es aber die höhere Schreibe/Lese-Geschwindigkeit von MongoDB, die für diese Datenbank gesprochen hat [16, 17].

MongoDB	CouchDB
+ Höhere Lese-Geschwindigkeit	+ Mobil-Support
+ Große User-Community	+ HTTP-Schnittstelle
+ Sharding	+ Sharding (seit neuerem)
	+ Master-Master Modus
+ Konsistenz	+ Verfügbarkeit
+ Vorhandene Erfahrung	

Tabelle 3 Argumente für die Datenbanken

4.3 Web-Framework

Als Programmiersprache wird Python gewählt, da mit den Bibliotheken pandas und seaborn ein mächtiges Datenanalyse- und Visualisierungstool zur Verfügung steht. Für die Web-Anwendung stehen in Python eine Reihe von Frameworks zur Verfügung. In den Fokus wurden die beiden beliebten Optionen genommen, Flask und Django

4.3.1 Framework 1: Flask

Flask ist ein Mikro-Framework und zielt darauf ab seinen Kern einfach, aber erweiterbar zu halten. Eine simple „Hello World“ Anwendung ist mit Flask in 5 Zeilen Code geschrieben. Flask zeigt eine flache Lernkurve und bietet einen einfachen Einstieg durch Tutorials und weiten Support in der Community.

Flask baut auf Werkzeug auf, welches das Standard Python Interface WSGI implementiert. Für das Frontend nutzt Flask Jinja, eine Template Engine, um die Seiten der Anwendung zu rendern [18]. So bietet der Kern einen Entwicklungs-Server mit Debugger, Unit Test-Support und RESTful request dispatching.
Flask bietet keine Built-in Funktionalitäten für Datenbankanbindung, Security und Validierung. Diese müssen vom Nutzer entweder selbst implementiert oder als Plugin von anderen dazu geholt werden, wobei das Flask Team solche Standard-Web-Anwendungs-Funktionalitäten als „approved extensions“ angibt.
Durch diesen simplen Aufbau und seiner Erweiterbarkeit bietet sich Flask insbesondere für kleinere Web-Anwendungen oder für Prototypen an [19], auch wenn dies keinen Abstrich für größere Projekte bedeutet.
Schlussendlich liegt es gänzlich in der Hand des Entwicklers, welche Anforderung seine Flask Anwendung am Ende erfüllen soll.

4.3.2 Framework 2: Django

Django baut auf einer abgewandelten Form des MVC Prinzipes auf.
Im Gegensatz zum Mikro-Framework Flask bietet Django von sich aus eine eigene Template-Sprache, ORM sowie ein Admin-Interface für das Applikationsdaten-Management. Out-of-the-box bietet Django außerdem Account Management und Session-Support über das Nutzer-Modell.
Das Framework liefert typische Funktionalitäten einer Web-Applikation von sich aus („Battery-included“) [20], gibt dabei dem Entwickler eine gewisse Doktrin vor. Für das ORM funktionieren standardmäßig relationale Datenbanken wie SQLite, PostgreSQL, MySQL, und Oracle optimal mit Django. Für andere oder nicht-relationale Datenbanken bedarf es entsprechender Anpassung seitens des Nutzers.

Vergleich

Flask eignet sich für eine kleine Webanwendung und dem Prototyping optimal und im Gegensatz zu Django muss ein geringerer Aufwand für die Anbindung an eine nicht-relationale Datenbank betrieben werden [19]. Ebenso passt die modulare Philosophie des Mikro-Frameworks besser in das Konzept eine kleinen, anpassbaren Analyse-Webanwendung.

Flask	Django
+ flexible Komponenten	+ Battery-Included (ORM, Template-System, Admin-Interface, …)
+ erweiterbar	+ keine weitere Abhängigkeiten
+ Minimaler Umfang	+ große User-Community
+ vorhandene Erfahrung	- unflexible mit NoSQL Datenbanken
- potenziell zeitaufwändiger	- Funktionsumfang zu groß für Anwendung
- externe Abhängigkeiten (Werkzeug, Jinja2)	

Tabelle 4 Vorteile und Nachteile der Web-Frameworks

☐ sekundäre Datenbankmodelle berücksichtigen — 47 Systeme im Ranking, März 2020

Rang Mär 2020	Rang Feb 2020	Rang Mär 2019	DBMS	Datenbankmodell	Punkte Mär 2020	Punkte Feb 2020	Punkte Mär 2019
1.	1.	1.	MongoDB	Document, Multi-Model	437,61	+4,28	+36,27
2.	2.	2.	Amazon DynamoDB	Multi-Model	62,51	+0,38	+8,02
3.	3.	3.	Couchbase	Document, Multi-Model	32,08	-0,08	-1,72
4.	4.	4.	Microsoft Azure Cosmos DB	Multi-Model	31,63	-0,32	+6,81
5.	5.	5.	CouchDB	Document	18,14	+0,00	-0,50
6.	6.	↑ 7.	Firebase Realtime Database	Document	12,60	+0,25	+2,30
7.	7.	↓ 6.	MarkLogic	Multi-Model	11,93	-0,32	-1,81
8.	8.	8.	Realm	Document	9,47	+0,62	+2,50

Abbildung 7 Ranking von dokumentenbasierten Datenbanken

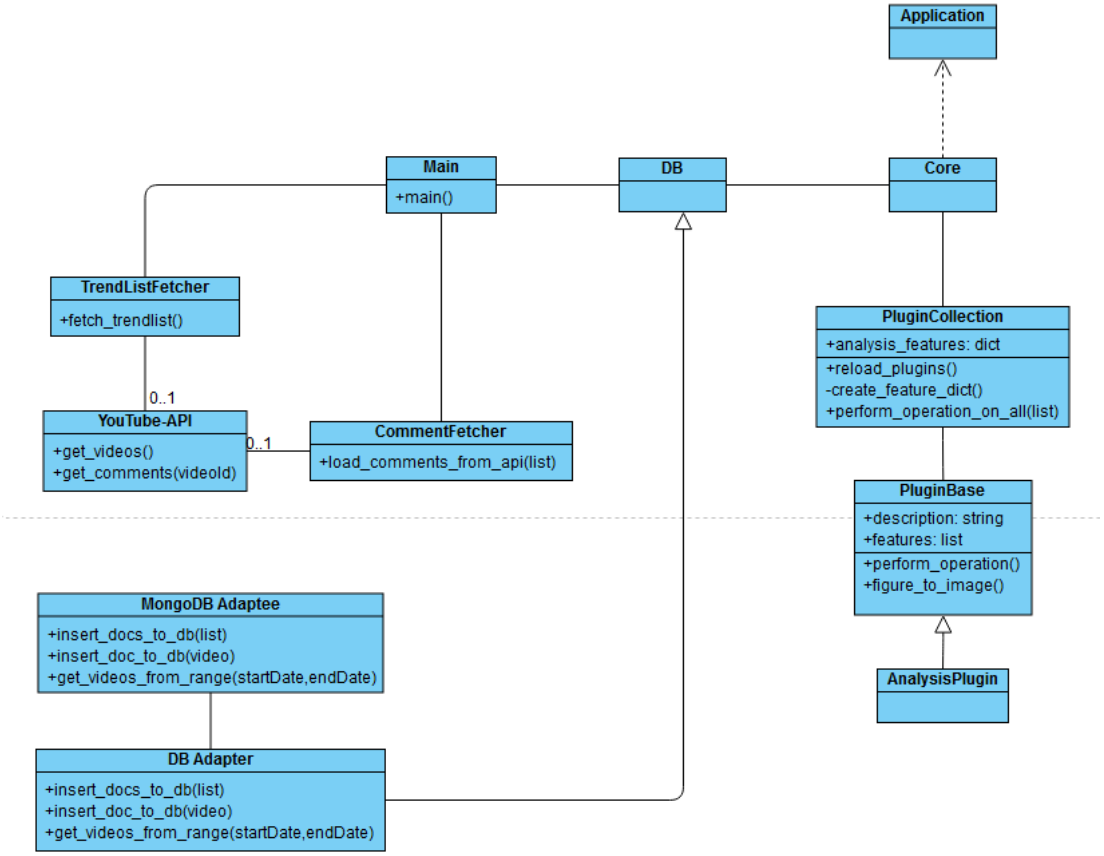

Abbildung 8 Klassendiagramm

5 Implementierung

Basierend auf den Entscheidungen aus dem vorherigem Konzept Kapitel wird nun die daraus resultierende Umsetzung und Implementierung in diesem Kapitel geschildert. Es werden grob einige nennenswerte Klassen und ihre Aufgaben angesprochen

5.1 YouTube API

Eine Klasse übernimmt die Abfragen an die API. Die Klasse fängt die Fehler seitens der API ab (API nicht verfügbar, Limit überschritten, …) und loggt diese.

Die YouTube API-Klasse übernimmt die Anbindung an die offizielle YouTube-API. Alle Anfragen erfolgen über diese Klasse und die eventuellen Fehler seitens der API (Nicht verfügbar, Limit überschritten) werden abgefangen und geloggt.
Die Methoden für Kommentar- und Video-Anfragen enthalten Parameter mit default Werten für die API.

5.1.1 Abfrage Frequenz

Für die Kommentar-Sektion werden die zu dem Zeitpunkt relevantesten Kommentare unter einem Video abgerufen.
Zunächst wurden 200 Kommentare pro Video als eine ausreichende Zahl angenommen um eine relevante Anzahl innerhalb der gegebenen Rahmenbedingungen zu haben.

> 200 Kommentare bedeuten 2 Requests mit Kosten von 6 Einheiten/Video.
> 50 Videos sind in der Liste = 300 Einheiten/Liste
> Geplante stündliche Wiederholung = 7200 Einheiten/Tag (+ 7 für die Liste)

Mit dieser Konfiguration stellt das tägliche Limit der API also kein weiteres Problem dar und es kann stündlich die Trendliste abgefragt werden.

5.2 Datenerfassung

Die Datenerfassung erfolgt über ein einfaches Skript in welchem die erforderlichen Klassen für API, Video-Liste-Verarbeitung und Kommentar-Verarbeitung initialisiert werden und die entsprechenden Schritte abgearbeitet werden (siehe Abbildung 9). Das Skript wird stündlich (siehe 5.1.1) von einem Cronjob angestoßen.

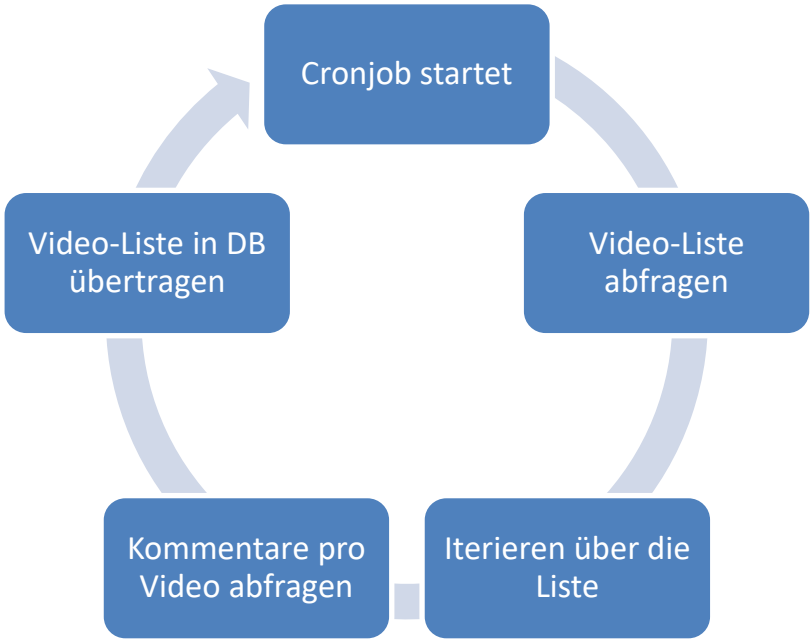

Abbildung 9 Datenerfassung-Ablauf

5.2.1 Video-Liste abfragen

Die TrendListFetcher-Klasse fordert über die API-Klasse die Video-Liste an und bereitet diese auf.

Mit dem Aufruf der fetch_trend_list Methode wird die Anfrage gestartet, wobei sich dies sich je nach Konfiguration der TrendListFetcher-Klasse unterscheiden kann.

Die Idee besteht darin, mehrere Möglichkeiten zu haben, um die Video-Liste abfragen zu können (siehe 4.1.1). So muss lediglich diese Klasse angepasst werden, wenn man über andere Wege diese oder ganz andere Listen abfragen möchte.

Die von der API zurückgegeben JSON wird noch einmal aufbereitet. So besteht die Ursprungs-JSON aus geschachtelten JSONs, welche die spätere Abfrage über die Datenbank minimal verkomplizieren.
Ebenso werden noch zusätzliche Informationen hinzugefügt, wie etwa der Listenabruf-Zeitstempel und die Platzierung des Videos in der Liste, um etwaige Reihenfolgestörungen vorzugreifen und Abfragen in diese Richtung zu vereinfachen.

```
{
  "kind": "youtube#videoListResponse",
  "etag": etag,
  "nextPageToken": string,
  "prevPageToken": string,
  "pageInfo": {
    "totalResults": integer,
    "resultsPerPage": integer
  },
  "items": [
    video Resource
  ]
}
```

Abbildung 10 Response JSON Schema

```
{
  "kind": youtube#video,
  "etag": etag,
  "id": string,
  "snippet": {
    "publishedAt": datetime,
    "channelId": string,
    "title": string,
    "description": string,
    "thumbnails": {
      (key): {
        "url": string,
        "width": 0,
        "height": 0
      }
    },
    "channelTitle": string,
    "tags": [
      string
    ],
    "categoryId": string,
    "liveBroadcastContent":
string,
    "defaultLanguage": string,
    "localized": {
      "title": string,
```

```
{
    "publishedAt": datetime,
    "channelId": string,
    "title": string,
    "description": string,
    "thumbnails": {
        (key): {
            "url": string,
            "width": integer,
            "height": integer
        }
    },
    "channelTitle": "Dude
Perfect",
    "tags": [
        string
    ],
    "categoryId": string,
    "liveBroadcastContent":
string,
    "localized": {
        "title": string,
        "description": string
    },
    "defaultAudioLanguage":
string,
```

<table>
<tr>
<td>

```
      "description": string
    },
    "defaultAudioLanguage":
string
  },
  "contentDetails": {
    "duration": string,
    "dimension": string,
    "definition": string,
    "caption": "string",
    "licensedContent": boolean,
    "regionRestriction": {
      "allowed": [
        string
      ],
      "blocked": [
        string
      ]
    },
  "statistics": {
    "viewCount": long,
    "likeCount": integer,
    "dislikeCount": integer,
    "favoriteCount": integer,
    "commentCount": integer
  }
}
```

</td>
<td>

```
    "id": string,
    "timestamp": datetime,
    "trendPlacement": integer,
    "duration": string,
    "dimension": string,
    "definition": string,
    "caption": boolean,
    "licensedContent": boolean,
    "projection": string,
    "durationInSec": integer,
    "viewCount": integer,
    "likeCount": integer,
    "dislikeCount": integer,
    "favoriteCount": integer,
    "commentCount": integer
}
```

</td>
</tr>
</table>

Tabelle 5 Antwort von API (links) vs. Bearbeitet (rechts)

Zusätzlich werden Features in die entsprechenden Datentypen konvertiert, da sie im Original JSON als Strings übergeben werden, entgegen dem, was YouTube angibt (siehe Abbildung 10).

5.2.2 Kommentare abfragen

Die Kommentare der Videos werden über die CommentFetcher-Klasse angefordert. Bei Initialisierung wird wie bei der TrendListFetcher ein API-Objekt übergeben, woraufhin mit der fetch_comments(list) Methode die Kommentare für jedes Video in der Liste angefordert wird.

Das von der API zurückgegebenen JSON werden ebenso wie das JSON der Trendliste geglättet und Datentypen entsprechend konvertiert.

5.3 Datenbank

Die Datenbank Anbindungsklassen (im Fall dieser Arbeit für MongoDB) implementieren das IDatabaseConnector Interface über welches sowohl das Datenerfassungsskript (siehe 5.2) als auch die Flask Anwendung (siehe 5.5.1) mit der Datenbank kommunizieren.
Eine Service Locator-Klasse “PluginCollection” sammelt alle Klassen in dem DB Ordner ein, die das gegebene Interface implementieren.
Die Auswahl der gewünschten Datenbank erfolgt dann über eine Konfigurationsdatei.

Aktuell nicht implementiert, aber ein interessantes Feature für die Zukunft, wäre die Wahl der Datenbank über die Weboberfläche.

5.4 Analyse Plugins

Bei den Analyse Plugins sammelt ebenso die “PluginCollection” jene Klassen ein, die hier eine Superklasse “PluginBase” implementieren.
Diese Superklasse bietet mit der Methode “figure_to_image”, eine Möglichkeit die Graphen der matplotlib Python-Bibliothek in PNG Bilder für das Frontend umzuwandeln.
Die Methode „perform_operation“ ist von allen erbenden Klassen zu implementieren. Diese wird vom Core über die PluginCollection aufgerufen, sobald eine Anfrage für eine Analyse eintrifft (siehe 5.5.3)

5.4.1 SentimentAnalysis

In den Analyse-Plugins, die sich mit den Kommentaren beschäftigen wird die Python-Implementierung von VADER konkret eingebunden. Die Kommentare werden zur Laufzeit auf ihr Sentiment untersucht. Dies hat den Hintergrund, dass im Falle des Zugangs zu einem „besseren“ Lexikon als dem aktuell gegebenen, dieses einfach getauscht werden kann (modularer Ansatz).
Alternativ bietet sich so auch die Möglichkeit Vergleiche unterschiedlicher SA Methodiken mit diesen Plugins vorzunehmen. Diese Implementierung bleibt jedoch zunächst bei dem ausgewählten Lexikon.

5.5 Web-Anwendung

Die Flask Anwendung besitzt in ihrem gegenwärtigem Zustand die einfache Funktion die entsprechende URL zu mappen und Anfragen an diese zu verarbeiten. Da der Hauptfokus mehr auf den vorher genannten Komponenten gesetzt wurde, beinhalten die folgenden Klassen keine komplexe Mechanismen und verlassen sich auf die Funktionalität ihres Frameworks.
Dieser Zustand ist mehr einem Prototyp gleich als einem produktivem System. Funktionalitäten wie Optimierung durch Caching oder Security wurden hier zunächst ausgelassen.

5.5.1 Flask-Applikation

In der application Klasse wird die Flask-Applikation initialisiert und die notwendige Konfiguration geladen.

5.5.2 Core-Klasse

Eine Core-Klasse, welche mit der Flask-Applikation initialisiert wird. Zum Zeitpunkt der Initialisierung dieser Klasse, werden die Service Locator angestoßen.
Die Core Klasse bietet somit die Schnittstelle zur Datenbank und den Analyse-Plugins für die Flask-Applikation.

5.5.3 Flask-Routen

Um die Hypothesen prüfen zu können, sind folgende Routen aktuell implementiert: Index und ein Auswertungsformular. Die Index-Seite leitet momentan zu dem Auswertungsformular weiter und gibt den Status der Datenbank wieder (verbunden/nicht-verbunden).
Auf dem Auswertungsformular (Abbildung 11) können gesammelte Videos innerhalb eines Zeitrahmens ausgelesen werden. Über die Feature-Wahl unterhalb können Auswertungen für diese angefertigt werden, sofern Analyse-Plugins dafür existieren. Auf der rechten Seite werden dann die Graphen und Tabellen von den Plugins angefertigt, wobei jedes Plugin seinen eigenen Tab für seine grafische Auswertung erhält.

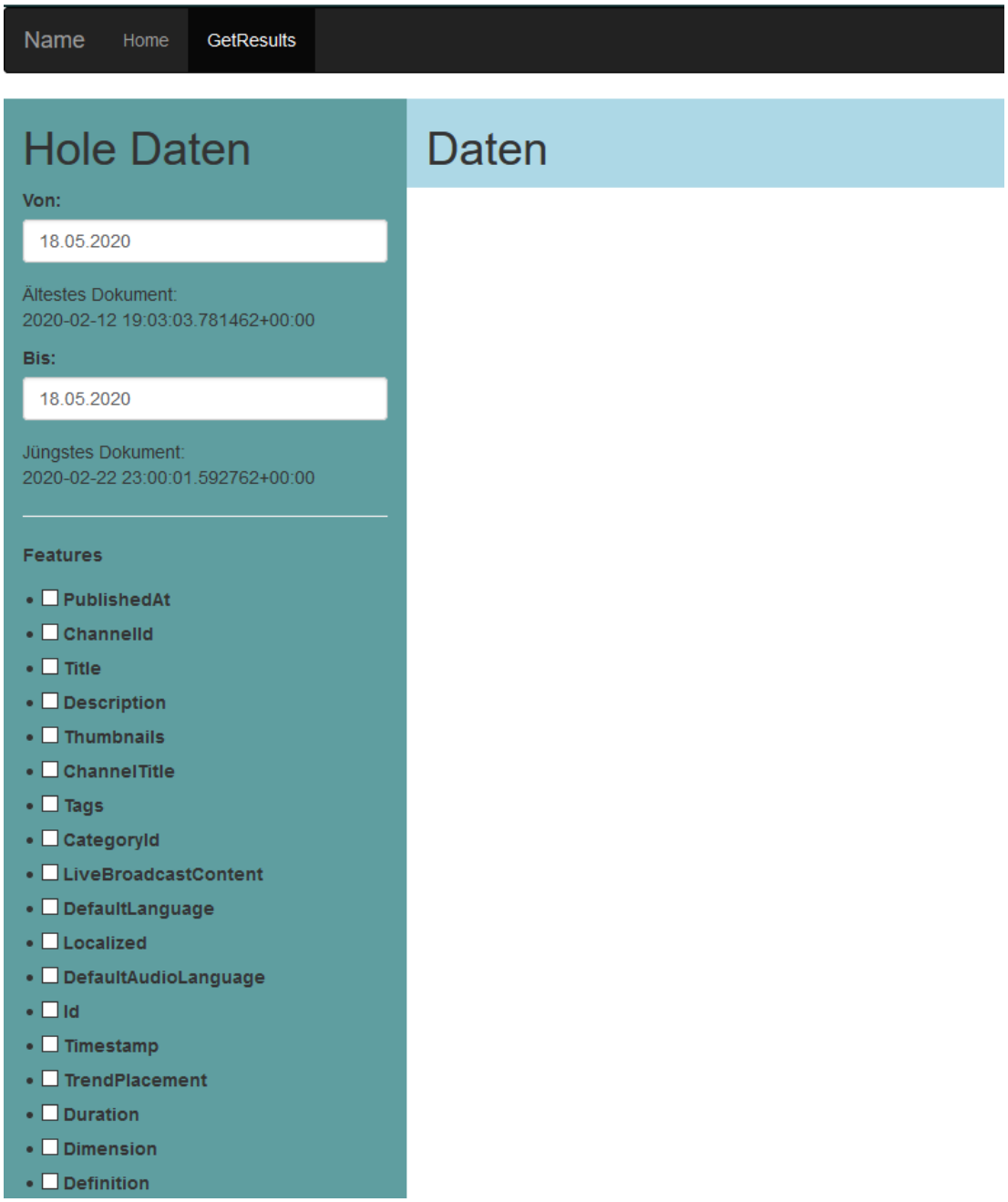

Abbildung 11 Datenanforderungsformular

6 Auswertung

In diesem Kapitel werden die Ergebnisse mithilfe der Implementierung des vorherigen Kapitels ausgewertet und die Hypothesen aus Kapitel 3 zu überprüft.
Es wurden über einen Zeitraum von 5 Monaten (18.02.20 – 16.07.20) wurden 167.238 Einträge in der Datenbank erfasst, wovon 5.191 (3,1% des gesamten Datensatzes) einzelne Videos darstellen. Alle Videos weisen eine Gesamtlaufzeit von 62 Tagen 21 Stunden auf. Stündlich kommen im Durchschnitt ≈ 1,4 neue Videos in die Trends.
An Kommentaren wurden insgesamt 37.000.000 Einträge erfasst, Duplikate inbegriffen. Von diesen Einträgen sind 7.289.701 Kommentare einzigartig, wobei 5.917.081 davon für die SA nutzbar sind. Diese sind entweder in englischer Sprache verfasst oder stellen verschiedene Emoticon-Kombinationen dar, die mit dem VADER SA-Lexikon ebenfalls auswertbar sind.

6.1 Trend-Akzeptanz

Die erste Auswertung beginnt damit, eine potenzielle Akzeptanz der Trend-Liste seitens der Nutzer zu ermitteln.
Zuallererst wurde die Hypothese aufgestellt, inwieweit Nutzer diese von YouTube erstellte Liste akzeptieren bzw. wie positiv sie die Videos in den Trends einschätzen, insbesondere im Bereich der Kommentarsektion über SA. Des Weiteren soll geschaut werden, ob sich ein Unterschied in der Bewertung (Likes/Dislikes u. Kommentare) erkennen lässt.

6.1.1 Herangehensweise

Hierfür werden zunächst die Werte aus den Video-Metadaten hinzugezogen. Mit den Likes und Dislikes wird ausgewertet wie die Nutzer, die mit diesen Trend-Videos interagiert haben, dieses einschätzen. Dazu wird die maximale Zahl der Likes und Dislikes eines Videos verwendet, was dem letzten Stand in der Liste entspricht.
Der nächste Schritt ist, dass die Nutzer-Akzeptanz, die über die Kommentarsektion mit dem Video interagiert haben, mittels Sentiment Analyse ausgewertet und mit den Werten aus dem voran gegangenen Schritt verglichen wird. Zuvor gilt es die nicht-englischsprachigen Kommentare aus dem Datensatz zu filtern, da deren Sentiment durch das Lexikon nicht ermittelt werden können. Ebenso werden jene betrachtet, die nur in Emojicons verfasst wurden, da diese von VADER ebenfalls analysiert werden können.

6.1.2 Ergebnisse & Interpretation

Insgesamt weisen die Trends eine eher positive Nutzer-Interaktion über die Like-Funktion auf. Im Durchschnitt liegt die Like-Rate bei 95,6%. Das Video mit der geringsten Akzeptanz lag bei 6%, die Höchste liegt bei 99,9%. Insgesamt betrachtet (Tabelle 6) liegt ein Großteil der Werte (75%) im Schnitt bei 98,8%. Auch im unteren Bereich (25%) ist der Schnitt über 90%.

	Like-Rate
mean	0,956
std	0,071
min	0,064
25%	0,958
50%	0,979
75%	0,988
max	0,999

Tabelle 6 Deskriptive Statistiken der Like-Rate

Der Anteil an schlecht bewerteter Videos ist gering. Von den 5.191 Videos haben 24 eine Like-Rate von 50% oder weniger, was 0,5% des Datensatzes entsprechen.
Die Videos, die in ihrer Like-Rate unter 25% liegen, sind nach dem Zeitraum der Datenerfassung nach Überprüfung alle gesperrt, auf privat gestellt oder sind anderweitig nicht mehr verfügbar.

Nach Auswertung der SA für die Kommentare zeigt sich, dass ein Großteil der Werte eher im positiven als im negativen Bereich liegt. Die positiven Kommentare liegen zu 43,8% vor. Der hohe Anteil an neutralen bzw. schwachen Sentiment-Werten (± 0.25) kann darin begründet liegen, dass VADER entweder nicht in der Lage war, den Text zu analysieren (was zu einer 0 im Composite führt) oder die Nutzer keine Schlagworte für eine der beiden Richtungen verwendet haben.

Sentiment Range	**Kommentarzahl**
(-1.0, -0.75]	248.551
(-0.75, -0.5]	435.147
(-0.5, -0.25]	495.457
(-0.25, 0.0]	1.864.429
(0.0, 0.25]	301.718
(0.25, 0.5]	818.976
(0.5, 0.75]	854.154
(0.75, 1.0]	898.643

Tabelle 7 Sentiment-Wertverteilung

Im direkten Vergleich geben beide Auswertung eine positive Interaktion der Nutzer mit den Trend-Videos wieder entgegen der zuvor aufgestellten Hypothese (siehe 3.2.1). Die Like-Rate

geht eindeutiger in die positivere Richtung im Vergleich zu den Kommentaren. Eine Ursache kann die Reichweite beider Werte sein, da für die SA diese größer (variable -1 – 1) und offener als die Like-Funktion (binäres Ja oder Nein) sind.

6.2 Trend-Aufbau

Nachdem die Nutzerakzeptanz der Trend-Liste ermittelt wurde, gilt es nun zu untersuchen, was YouTube für Videos in seine Liste aufnimmt und welche Kriterien sie dazu erfüllen müssen.
In den Grundlagen (siehe 2.1) wurde bereits eine Liste mit grobe Kriterien von YouTube aufgestellt. Da YouTube bewusst nicht weiter auf Details einzugehen scheint, wird nun versucht anhand der erfasst Video-Metadaten eigene Kriterien für ein Trendvideo auszumachen. Gleichzeitig wird geprüft, inwieweit sich YouTubes Kriterien numerisch beschreiben lassen (siehe 3.2.2).

Hier noch einmal die YouTube-Kriterien:

1. Anzahl der Aufrufe
2. Wie schnell ein Video Aufrufe generiert
3. Quellen der Aufrufe (auch außerhalb von YouTube)
4. Alter des Videos
5. Leistung des Videos im Vergleich zu anderen Uploads auf demselben Kanal

6.2.1 Herangehensweise

Punkt 3 und 5 liegen außerhalb des Rahmens dieser Arbeit und des Datensatzes, weshalb diese Punkte nicht überprüft werden. Allerdings stellt insbesondere Punkt 3 mit „Quellen der Aufrufe außerhalb von YouTube“ Potenzial für zukünftige Arbeiten dar.
Somit folgen in diesem Abschnitt die Überprüfung der Punkte 1, 2 und 4. Hierfür wird der Stand der Daten eines Videos genommen, wie er das erste Mal von der API abrufbar gewesen ist (folgend als Erstplatzierung beschrieben). Weitere Entwicklung dieser Werte während sich ein Video in den Trends befindet und evtl. die Platzierung beeinflussen werden in einem separaten Abschnitt vorgenommen.
Ergänzend zu dieser Auswertung wird noch die Zusammensetzung der Trend-Liste anhand weiterer Metadaten betrachtet. So werden noch zusätzliche Features wie Laufzeit und Kategorie in die Betrachtung hineinbezogen, um mögliche weitere Kriterien zu ermitteln.

6.2.2 Ergebnisse und Interpretationen

Für Punkt 1 – Anzahl der Aufrufe – zeigt sich, dass im Durchschnitt ein Video 875.646 Views hat, wobei 50% des Datensatzes bei 415.902 Views im Schnitt liegen (Tabelle 8). Für eine Platzierung in den Trends sind etwa 400.000 bis 875.000 Views notwendig anhand der vorliegenden Daten. Beim Alter liegen Videos bei knapp Ø 18 Std, wohin gegen die Extreme

bei 1,25 Std (min) und 25 Tagen (max) stehen (Tabelle 9). Orientiert man sich hier auf den Schnitt bei 50% und dem Durchschnittswert lässt sich für das Alterskriterium eine Spanne von 16,5 – 18 Std ermitteln, in dem ein Video für eine Erstplatzierung mit seinem Alter liegen sollte. Anhand dieser Werte lässt sich eine Annahme für die View-Anzahl pro Stunde treffen, die ein Video womöglich im Optimum braucht: 475.000 Views Ø / 17 Std (Alter Ø) = 27.941 Views/Std

	Views
mean	875.646
std	2.315.047
min	30.927
25%	217.145
50%	415.902
75%	818.922
max	58.362.006

Tabelle 8 Deskriptive Statistiken für Views

	Video Alter
mean	0 days 17:55:38
std	0 days 17:40:56
min	0 days 01:15:53
25%	0 days 09:56:18
50%	0 days 16:24:22
75%	0 days 21:55:32
max	25 days 14:59:59

Tabelle 9 Deskriptive Statistiken für Video-Alter

YouTube bildet also mit seiner Trendliste nicht ganz das aktuelle Weltgeschehen ab, da sie in der Regel fast einen Tag zurück hängen, bis auf einige Ausnahmen. Eine Möglichkeit könnte das Kriterium „View über Zeit“ (Punkt 2 der YouTube Kriterien) darstellen. Das jüngste Video (1,25 Std alt) im Datensatz hat er in der Zeit ≈10.000.000 Views angesammelt, das Video mit den meisten Views (≈58.000.000) brauchte 2 Tage, bevor es in der Liste angezeigt wurde. Wenn gleich die Gesamtaufrufzahl ein Kriterium darstellt, scheint es eine niedere Priorität zu haben als das „View über Zeit“ Kriterium.

Betreffend der Laufzeit wurden anfangs Annahmen getroffen, dass Videos eine geringe Laufzeit (≈ 3,5 min) aufweisen würden. Die anfängliche Annahme beruhte auf erste Beobachtungen zu Beginn der Datenerhebung. Jedoch liegt ein geringerer Anteil der Videos in diesem Bereich (25% der Daten) (Tabelle 10). Der Großteil (75%) hat eine Laufzeit von 15,5 Min, wohingegen der Durchschnitt bei 17,4 Min liegt. Die Laufzeit der Videos orientiert sich offensichtlich über der Minimumlaufzeit, die ein Video aufweisen muss, um mit der maximalen Anzahl von Werbeclips versehen werden zu können.

	Laufzeit in Min
mean	17,45
std	43,60
min	0,00
25%	3,52
50%	8,82
75%	15,53
max	715,02

Tabelle 10 Deskriptive Statistiken für Laufzeit

Die beiden Kategorien „Entertainment“ (*1.119 ≅ 21,56%*) und „Music“ (*988 ≅ 19,03%*) machen einen Großteil der Videos in den Trends aus (Abbildung 15). Musik-Videos weisen eine geringere Laufzeit auf als Entertainment-Videos (Abbildung 12). Videos, die in den Aktivismus-Bereich fallen, haben im Schnitt eine fast viermal so längere Laufzeit als „Travel & Events“-Videos, wobei beide Kategorien in ihrer Zahl kaum in den Trends vertreten sind.
Die Entertainment-Kategorie stellt eine Art Universal-Kategorie dar, in welcher Content-Creator im Zweifel ihre Werke kategorisieren können. Für eine allgemeine Trendliste verwundert es daher nicht, wenn die Universal-Kategorie überwiegend vertreten ist. Der hohe Musik-Video Anteil rührt daher, dass YouTube neben dem Unterhaltungsfaktor auch eine große Plattform für Musikvideos ist. Obwohl diese Kategorien diejenigen sind, die am meisten in den Trends vertreten sind, sind sie nicht die Meistgeschauten.
Im Median schafft es Entertainment noch auf Platz 3 (Abbildung 13), während „Science & Technology“ die meisten Views aufweisen. Jedoch kann keine pauschale Aussage getroffen werden, ob eine Kategorie von Nutzern eher geschaut wird als eine andere, da diese Daten lediglich eine Momentaufnahme darstellen. Für den beobachten Zeitraum lässt sich sagen, dass Videos aus dieser Kategorie „sehenswerter“ waren als die anderen aus der Liste. Die Trends sind dafür zu abhängig von dem Geschehen in der Welt. Dasselbe gilt allerdings auch für die anderen Kategorie betreffenden Statistiken.

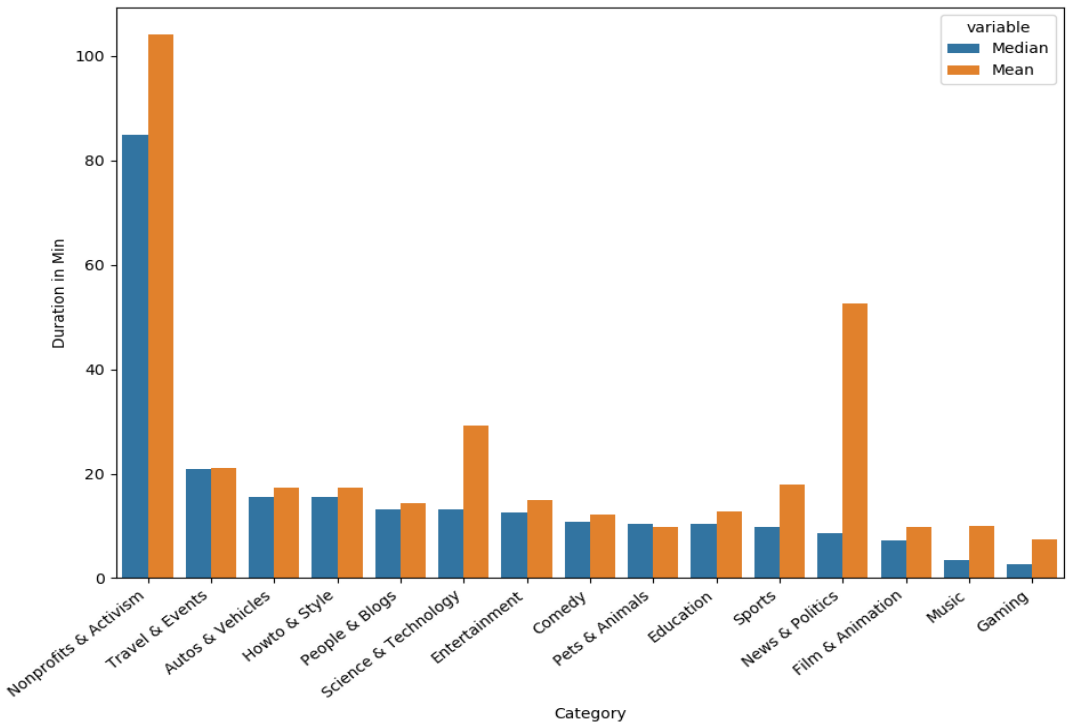

Abbildung 12 Durchschnittliche Laufzeit der Videos pro Kategorie

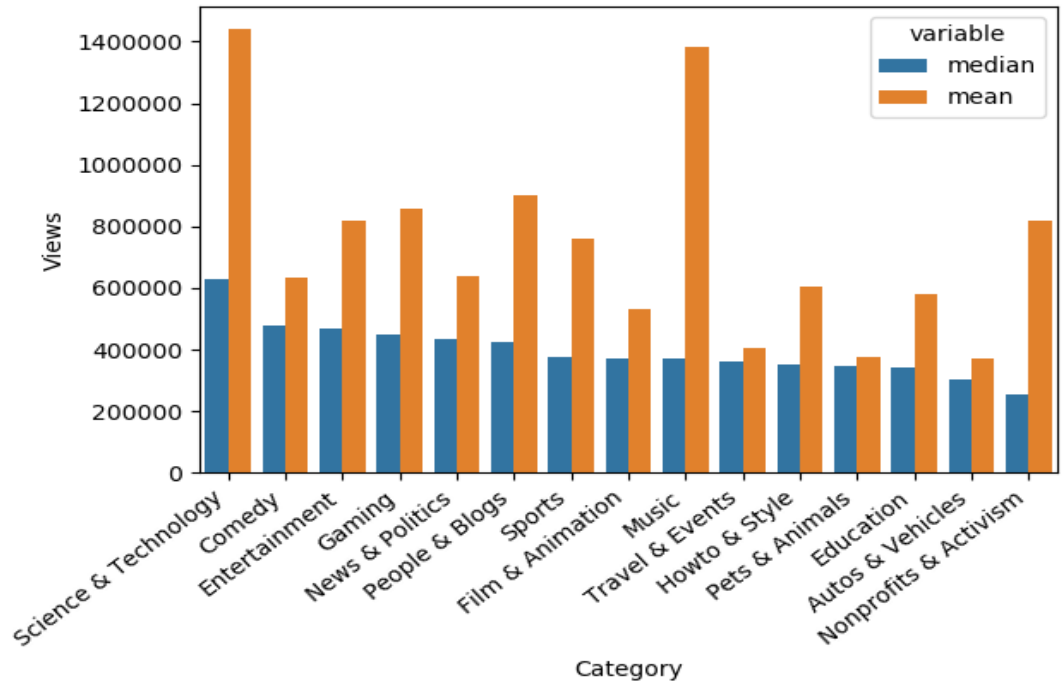

Abbildung 13 Views pro Kategorie

Bezüglich der Platzierung sind Videos in der Regel höher eingestuft (Ø Platz 20). Die meisten Erstplatzierungen liegen um Platz 10 herum (Abbildung 14). Insgesamt lässt sich beobachten, dass es mehr Videos gibt, die hoch einsteigen und womöglich eher über die Zeit in der Platzierung absteigen, als Videos, die auf niedrigeren Plätzen in die Trends kommen und sich hocharbeiten müssten.

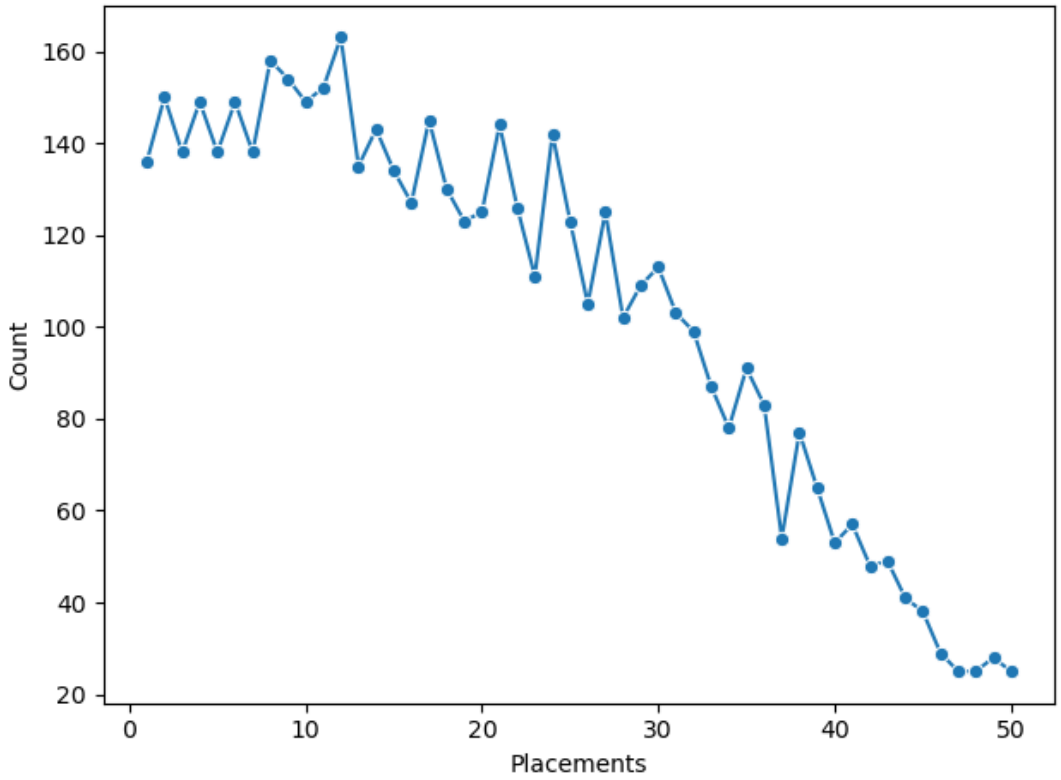

Abbildung 14 Video-Anzahl pro Trend-Platz (Erstplatzierungen)

Nennenswerte Erstplatzierungen:
87-ZFjLfBAQ auf Platz 1 mit einer Laufzeit von 0 Sek. und 1,5 Millionen Views. Hierbei handelt es sich um eine aktive Live-Übertragung, die es in die Trends geschafft hat. Es ist die einzige *laufende* Live-Übertragung im Datensatz.
GNEPwwv56DY (10. P), ***Pntmw76eQBc*** (30. P), ***fMSezPwq2js*** (1. P) und ***gLM1-M0oalg*** (9 P.) sind alles Videos von Google (GoogleDoodles) oder YouTube (YouTube Live Originals) in den Trends, welche keine Zähler aufweisen (Views, Likes und Dislikes) und deren Kommentarfunktion deaktiviert sind.

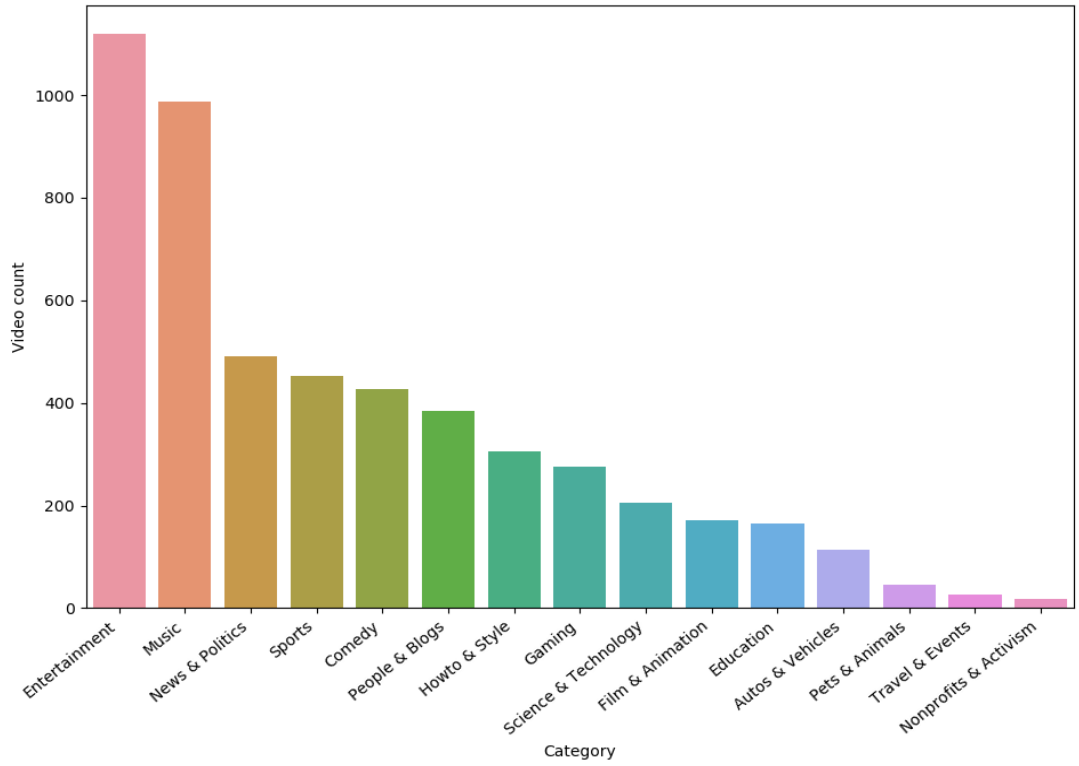

Abbildung 15 Anzahl Videos pro Kategorie

Kategorie	Anzahl	Anteil (%)
Entertainment	1119	21,557
Music	988	19,033
News & Politics	492	9,478
Sports	452	8,707
Comedy	427	8,226
People & Blogs	384	7,397
Howto & Style	306	5,895
Gaming	276	5,317
Science & Technology	205	3,949
Film & Animation	172	3,313
Education	166	3,198
Autos & Vehicles	113	2,177
Pets & Animals	45	0,867
Travel & Events	27	0,52
Nonprofits & Activism	19	0,366

Tabelle 11 Videoverteilung pro Kategorie

Kategorie	Platzierung (Median)	Platzierung (Ø)
Film & Animation	18	20
Autos & Vehicles	27	27
Music	19	20
Pets & Animals	20	22
Sports	17	19
Travel & Events	23	25
Gaming	14	16
People & Blogs	16	18
Comedy	18	20
Entertainment	19	20
News & Politics	15	17
Howto & Style	22	24
Education	25	25
Science & Technology	25	24
Nonprofits & Activism	13	13

Tabelle 12 Durchschnittsplatzierung pro Kategorie

6.3 Wiederkehrende Kanäle

Bedingt durch den Grundsatz „Quantität über Qualität“ besteht die Hypothese, dass Content-Creator in einer hohen Frequenz ihre Videos auf YouTube hochladen, bevorzugt vom YouTubes Algorithmus berücksichtigt und den Nutzern angezeigt zu werden. Es besteht die Erwartung, dass mit einer hohen Zahl Kanäle mehr als 1 oder 2 verschiedene Videos auf einem Platz in der Liste hatte.

6.3.1 Herangehensweise

Es wurden hierfür alle Kanäle gruppiert, die dazu assoziierten, einzigartigen Video-IDs gesammelt und ihre Anzahl ermittelt. Zusätzlich wurden die Laufzeit und die Durchschnittsplatzierung aller Videos der Kanäle und deren Durchschnittsplatzierung in den Trends zu der Auswertung hinzugezogen.

6.3.2 Ergebnisse und Interpretationen

Im Durchschnitt hat ein Kanal rund 3 Videos in den Trends. Insgesamt haben 1.870 unterschiedliche Kanäle mindestens ein Video in den Trends gehabt, wovon die Hälfte (51,8%) nur einmalig in den Trends vertreten sind. Die Kanäle mit den meisten Videos (Tabelle 13) stellen eine extreme Ausnahme dar, wobei knapp 4% mit mehr als 10 Videos in den Trends vertreten waren (Tabelle 14).

#	Kanalname	Videozahl	Kategorien
1.	NBC News	47	News&Politics
2.	Linus Tech Tips	34	Science&Technology
3.	CNN	33	News&Politics
4.	ESPN	31	Sports
5.	SSSniperWolf	31	Entertainment
6.	Good Mythical MORE	29	Entertainment
7.	Good Mythical Morning	28	Entertainment
8.	Skip and Shannon: UNDISPUTED	26	Sports
9.	The Try Guys	26	Comedy
10.	BBC News	25	News&Politics
11.	WWE	24	Sports
12.	Bon Appétit	24	Entertainment, Howto&Style

13.	The ACE Family	23	People&Blogs
14.	NBA	22	Sports
15.	James Charles	20	Entertainment

Tabelle 13 mehrfach vorkommende YouTube-Kanäle

Betrachtet man die Platzierungen in den Trends, so liegen die Top 15 Kanäle mit ihre durchschnittlichen Platzierung um die allgemeine Durchschnittsplatzierung (siehe 6.2), bis auf wenige Ausnahmen (Platz 2, 5 oder 9). Die Laufzeit liegt bei um die 10 Minuten, wobei dies mit der Minimallaufzeit für maximale Werbung (wie bereits in 6.2 erläutert) zusammenhängt. Einzig drei davon liegen darunter.

#	Kanalname	Trendplatzierung Ø	Laufzeit Ø
1.	NBC News	23	3603
2.	Linus Tech Tips	40	871
3.	CNN	22	615
4.	ESPN	19	431
5.	SSSniperWolf	35	630
6.	Good Mythical MORE	26	809
7.	Good Mythical Morning	27	895
8.	Skip and Shannon: UNDISPUTED	22	758
9.	The Try Guys	31	1223
10.	BBC News	25	638
11.	WWE	26	296
12.	Bon Appétit	28	1432
13.	The ACE Family	25	970
14.	NBA	27	584
15.	James Charles	28	913

Tabelle 14 Durchschnittswerte aller Videos pro Kanal

Videozahl in Trends	Kanäle	Anteil %
1	959	51.28
2	349	18.66
3	152	8.13
4	113	6.04
5	64	3.42
6	61	3.26
7	44	2.35
8	33	1.76
9	19	1.02
>10	76	4.06

Tabelle 15 Kanalanzahl mit mehreren Videos

Auffallend ist, dass viele Kanäle einiger bekannterer amerikanischer Fernsehsender in den Trends zu finden sind. Betrachtet man noch die zugeordneten Kategorien dazu, finden sich öfter „News“ und „Sport“ auf den oberen Plätzen neben der insgesamt breitvertretenen „Entertainment“ Kategorie.
Dass insbesondere Nachrichten-Sender mit einer höheren Uploadfrequenz in den Trends zu finden sind, im Gegensatz zu kreativschaffenden Content Creator, zeigt, dass YouTube einen höheren Fokus auf Quantität für die Trends zu legen scheint. Quantität bezieht sich hierbei auf die Laufzeit wie auch die Anzahl der Videos. Ergänzend kommt noch hinzu, dass in dem Zeitraum der Datenerhebung einige Ereignisse gefallen sind, die eine hohe mediale Aufmerksamkeit erzeugt haben. Die weltweite Covid-19 Pandemie, Protestunruhen und die im Herbst anstehenden US-Wahlen sind hier nur als Beispiele zu nennen.

6.4 Kommentar-Zusammensetzung

In diesem Abschnitt gilt es die Auswertung mittels der Kategorisierung, wie Schultes sie formuliert hat, durchzuführen. Einige der original formulierten Features, die hierfür angewendet werden, wurden teilweise abgeändert.
Gemäß der Arbeit von Schultes (siehe 2.2.2**Fehler! Verweisquelle konnte nicht gefunden werden.**) wird hier keine große Vorannahme getroffen für diese Hypothese. Schultes hatte angenommen, dass ein höherer Anteil von substanzlosen Kommentaren zu erwarten sei, durch welchen die Nutzer ein negatives Bild der Plattform haben sollten. Jedoch zeigte sich, dass keine der aufgestellten Kategorien überproportional vertreten war und der Anteil an substanziellen, mit den Video interagierenden Kommentaren sogar eher größer als die angenommene Kategorie. Es soll geschaut werden, inwieweit die Kommentare für diesen Datensatz in Schultes Kategorien eingeordnet sind, um anschließend eine mögliche Erklärung für dieses Resultat zu finden.

6.4.1 Herangehensweise

Zunächst wurden die Kommentare, wie auch für die Sentiment Analysis in der vorangegangenen Auswertung, die nicht in Englisch verfasst wurden, aussortiert. Anschließend werden Stoppwörter („a“, „the“, „he“, etc.) aus den Übrigen herausgefiltert, damit schlussendlich Schultes Features angewendet werden können.
Die Kategorie T1 (Diskussion) fällt aus der Betrachtung heraus, da Kommentar-Threads durch die gegebenen Begrenzung der API (siehe 4.1) nicht erfasst werden. Somit liegt der Fokus mehr auf den T2 (Substanzlos) und T3 (Substanziell) Kommentaren. Für die C5 Unterkategorie, wurde im Original als Feature „Emotional“ als Kriterium formuliert. Schultes hat dafür eine eigens zusammengestellte Liste angewendet, die bestimmte emotionale Schlagwörter enthielt für unterschiedliche Stimmungen. Für diese Arbeit wurde das Feature ausgelassen.

6.4.2 Ergebnisse und Interpretation

Nach Auswertung der einzelnen Features ergibt sich ein minimaler Abstand der Substanziellen Kommentaren zu den Substanzlosen.

Kategorie	Unterkategorie	Anzahl	%	Gesamt	%
Substanzlose	C4	248.857	4,21	1.934.688	32,70
Kommentare	C5	746.856	12,62		
(T2)	C6	938.975	15,87		
Substanzielle	C7	164.040	2,77	2.166.335	36,61
Kommentare	C8	443.621	7,50		
(T3)	C9	1.558.674	26,34		
Andere	C10	3.006.835	50,82	3.006.835	30,69

Tabelle 16 Kategorie-Einteilungen

Ähnlich wie in Schultes Arbeit zeigt sich für diesen Datensatz eine gleiche Verteilung auf die einzelnen Kategorien. Ledig die substanzvollen Kommentare zeigen einen minimalen Vorsprung zu den negativ aufgefassten Substanzlosen. Schultes beschrieb sein Ergebnis damit, dass YouTube Kommentare Real-Life Kommunikation abzubilden scheinen.

6.5 Auswirkung Video-Platzierung

Für diese Auswertung soll eine potenzielle Auswirkung einer Platzierung in den Trends untersucht werden. Die Annahme besteht, dass zumindest eine längere Position in der Liste – zunächst unabhängig, ob hoch oder niedrig – eine höhere Zahl an Nutzerinteraktion

bewirkt, wodurch ein indirekt überprüft werden kann, inwieweit Nutzer diese Liste aktiv nutzen.

6.5.1 Herangehensweise

Alle numerischen Metadaten eines Videos (Views, Likes, Dislikes, Kommentarzahl) werden über Zeit betrachtet, ihr Wachstum über die Zeit gemittelt und aus diesen errechneten Mittelwerten ein Gesamtwachstum ermittelt.

6.5.2 Ergebnisse und Interpretation

In der Betrachtung über Zeit wird jeder Eintrag eines Videos im Datensatz betrachtet. Im Durchschnitt befand sich ein Video für 32,2 Std (28 Std im Median) auf einer Trendplatzierung. 5 Videos schafften es dabei mehr als 120-mal in den Trends platziert zu sein, welche hierbei die Ausnahme darstellen. 63 Videos blieben bei einer Erstplatzierung und sind somit für eine weitere Betrachtung nicht von Relevanz.

Video ID	Titel	Kategorie	Vorkommen
4kJc-s4W6LU	Learning Tik Tok Dances From Larray & Addison Rae	Comedy	132
W2hRTTTpmr8	I Tried Following A Soap Cupcake Tutorial	People & Blogs	128
YV0Pa1cVvMk	TURNING THE DOLAN TWINS INTO TIKTOKERS FT. Addison Rae	People & Blogs	127
_sGlbF36L4g	My Dad is Gay | The Secret Life of Lele Pons	Entertainment	124
y8AOvb-Iy60	KSI – Cap (feat. Offset) [Official Music Video]	Music	121

Tabelle 17 mehrfach Platzierungen von Videos

Nicht zu erwarten waren die negativen Werte für die verschiedenen Werte (Views, Likes, Dislikes) (Tabelle 18). Ein Video hat im Laufe der Zeit, in welcher es in den Trends war, an Aufrufen verloren, statt welche zu generieren. Nach weiterer Recherche zeigt sich, dass YouTube „kontinuierlich die Gültigkeit von Aufrufen“ prüft und die Anzahl der Aufrufe „jederzeit angepasst werden“ kann [21]. Somit sind zumindest die Views von diesem Ausreißer aus YouTubes Sicht nicht rechtmäßig angesammelt worden und die Views wurden wieder abgezogen, wodurch diese negative Entwicklung zustande kam (gleiches ist für die Likes und Dislikes anzunehmen). Das entsprechende Video ist auch nicht mehr öffentlich sichtbar (Stand: 13.09.2020).

Für eine Erstplatzierung in den Trends wurde ein etwaiges Wachstum von 27.941 Views/Std ermittelt. Geht man von einer Auswirkung durch die Trend-Liste aus, so müsste also das View-Wachstum über den für die Erstplatzierung ermittelten Wert liegen. Während ihrer Zeit in der Liste haben Videos jedoch rund 27.000 Views/Std an Zuwachs gehabt. Somit ist zumindest im Mittel kein direkter Einfluss der Liste auf die Aufrufe zu erkennen.

#	Views Ø	Likes Ø	Dislikes Ø	Kommentare Ø
mean	26.269,05	934,46	40,55	83,26
std	2.626,91	2.829,22	161,73	351,08
min	-20.011,7	-3562,00	-138,00	-369,97
25%	6.009,04	129,25	5,71	13,1
50%	12.102,92	335,26	12,42	30,25
75%	26.492,45	804,42	30,44	68,28
max	149.201	81.655,04	4.944,73	12.783,54

Tabelle 18 Deskriptive Statistik Videometriken über Zeit

7 Zusammenfassung und Ausblick

7.1 Zusammenfassung

Die YouTube Trends sollen laut YouTube einen Einblick in das aktuelle Weltgeschehen geben und eine Diversität an Inhalten beinhalten. Nutzer, die mit den Trends interagieren, zeigen eine positive Resonanz mit den Inhalten der Liste, sowohl in ihrer binären Bewertung als auch in der direkten Interaktion mit dem Video via Kommentarsektion.
Allerdings sind diese Ergebnisse nicht fundiert genug, um eine konkrete Aussage über die allgemeine Akzeptanz dieses Angebotes wieder zu geben. Die Videos in der Liste sind für sich positiv bewertet worden.
Jedoch steht zur Debatte, ob diese Bewertung zustande kommt, weil ein Video in den Trends gelistet worden ist. Beliebte Videos werden den Nutzern über den YouTube Algorithmus direkt auf der Startseite vorgeschlagen unabhängig der Trendliste. Ein Vermerk auf eine Trendplatzierung ist dann unter dem Titel des Videos zu finden, sobald ein Nutzer sich ein solches Video anschaut. Somit ist fraglich, inwieweit Nutzer die Liste nutzen um beliebte Videos schauen.
Die hier vorgestellte Software bietet einen Einblick in einen Ausschnitt der YouTube Trends. Jedoch zeigten die abschließenden Analysen, dass noch einige Verbesserungen vorgenommen werden müssen, um mit der Gesamtheit der Daten umgehen zu können, die sich über mehrere Monate spannt. Insbesondere Pandas, eigentlich ein ideales Datenanalyse-Tool, zeigt Schwächen im Umgang mit mehreren Millionen Spalten.

7.2 Ausblick

Nach gegenwärtigem Stand ist fraglich, ob die Trendliste einen Einfluss auf die gelisteten Videos in ihren numerischen Metadaten haben. Nachfolgende Arbeiten könnten mittels Nutzerbefragungen und weiterer Auswertung mögliche Erkenntnisse liefern.
In den Grundlagen wurde des Weiteren ein Kriterium mit dem Einfluss eines Videos außerhalb von YouTube benannt, welches ebenso aufschlussreiche Auswertungen bieten könnte. Anfangs war auch eine Untersuchung der Kommunikation von Nutzern

untereinander über verschiedene Videos hinweg angedacht, welche jedoch angesichts der Limitierungen (API bspw.) nicht mehr zeitlich umzusetzen war.

Literaturverzeichnis

[1] Schultes, P., Dorner, V., & Lehner, F. (2013). Leave a Comment! An In-Depth Analysis of User Comments on YouTube. Wirtschaftsinformatik, 42, 659-673.

[2] statista, „statista," 2019 09 03. [Online]. Available: https://de.statista.com/themen/162/youtube/. [Zugriff am 2020 03 19].

[3] Google, „Google-Hilfe," 23 12 2019. [Online]. Available: https://support.google.com/youtube/answer/7239739?hl=de. [Zugriff am 10 03 2020]

[4] van Aken, B., Risch, J., Krestel, R., & Löser, A. (2018). Challenges for toxic comment classification: An in-depth error analysis. arXiv preprint arXiv:1809.07572.

[5] Read, J., & Carroll, J. (2009, November). Weakly supervised techniques for domain-independent sentiment classification. In *Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion* (pp. 45-52).

[6] Walaa Medhat, Ahmed Hassan, Hoda Korashy (2014). Sentiment analysis algorithms and applications: A survey. Ain Shams Engineering Journal. Volume 5, Issue 4. https://doi.org/10.1016/j.asej.2014.04.011.

[7] Thelwall, M. (2018). Social media analytics for YouTube comments: Potential and limitations. International Journal of Social Research Methodology, 21(3), 303-316.

[8] Thelwall, M. (2017). The Heart and soul of the web? Sentiment strength detection in the social web with SentiStrength. In *Cyberemotions* (pp. 119-134). Springer, Cham.

[9] Gatti, L., Guerini, M., & Turchi, M. (2016). SentiWords: Deriving a high precision and high coverage lexicon for sentiment analysis. IEEE Transactions on Affective Computing, 7(4), 409-421.

[10] Hutto, C.J. & Gilbert, Eric. (2015). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Proceedings of the 8th International Conference on Weblogs and Social Media, ICWSM 2014.

[11] Beasley, A., & Mason, W. (2015, June). Emotional states vs. emotional words in social media. In Proceedings of the ACM Web Science Conference (pp. 1-10).

[12] Ribeiro, F. N., Araújo, M., Gonçalves, P., Gonçalves, M. A., & Benevenuto, F. (2016). Sentibench-a benchmark comparison of state-of-the-practice sentiment analysis methods. *EPJ Data Science*, *5*(1), 1-29.

[13] „DB-Engines," [Online]. Available: https://db-engines.com/de/system/Terrastore. [Zugriff am 10 03 2020].

[14] Henricsson, R. (2011). Document Oriented NoSQL Databases: A comparison of performance in MongoDB and CouchDB using a Python interface.

[15] MongoDB, I. (2014). Mongodb. *URL https://www. mongodb. com/. [Zugriff am 09 04 2020]*.

[16] Bhardwaj, N. D. (2016). Comparative study of couchdb and mongodb–nosql document oriented databases. International Journal of Computer Applications, 136(3), 24-26.

[17] Henricsson, R. (2011). Document Oriented NoSQL Databases: A comparison of performance in MongoDB and CouchDB using a Python interface.

[18] Team, P. (2010). Flask Documentation.

[19] Ghimire, D. (2020). Comparative study on Python web frameworks: Flask and Django.

[20] Team, D. (2013). Django documentation.

[21] Google, „Google-Hilfe,“ 23 12 2019. [Online]. Available: https://support.google.com/youtube/answer/2991785?hl=de [Zugriff am 20 08 2020]

Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, den _____ _____