University of Applied Sciences
HOCHSCHULE EMDEN·LEER

Fachbereich Technik
Abteilung Elektrotechnik und Informatik

Data Mining mit Twitter

Entwicklung eines prototypischen Python Notebooks zur exemplarischen Extraktion und Auswertung von Daten des Social-Media-Dienstes Twitter anhand von Suchbegriffen oder einem Nutzernamen unter Nutzung der von Twitter zur Verfügung gestellten APIs, allgemein zugänglicher Bibliotheken sowie der Berücksichtigung des Open Source Frameworks TensorFlow

Christian Kitte
(Matrikelnummer xxxxxxx)

Erstprüfer	:	Dipl. Inform. Andreas Wilkens
Zweitprüfer	:	Dipl. Inform. Robert Bozic
Eingereicht am	:	31. Januar 2019

Abstract

Im Rahmen dieser Arbeit erfolgt die konzeptionelle Planung und prototypische Umsetzung eines auf Google Drive gehosteten Python Notebooks zur Abfrage von Daten des Dienstes Twitter anhand von Suchbegriffen oder eines Nutzernamens. Als Notebook kommt hierbei das frei zugängliche Colaboratory zum Einsatz.

In dem so erstellten Notebook wird exemplarisch das Vorgehen vom Erhalt der Daten über deren Normierung und semantischen Analyse bis hin zur Visualisierung des Ergebnisses ausführlich beschrieben und durchgeführt. Zum Einsatz kommen hierbei etablierte Bibliotheken wie das Natural Language Toolkit oder das Open Source Framework TensorFlow.

Dem Nutzer wird so ermöglicht, anhand einer einfach zugänglichen Anwendung diese Schritte konkret zu verfolgen und darüber hinaus aktiv im Code einzugreifen und ihn dadurch besser zu verstehen.

The conceptual planning and prototypical implementation of a Python notebook hosted on Google Drive for querying data of the Twitter service on the basis of search terms or a user name takes place within the scope of this work. The notebook used is the freely accessible Colaboratory.

In the so created notebook the procedure from the receipt of the data over their standardization and semantic analysis up to the visualization of the result is described in detail and accomplished exemplarily. Established libraries such as the Natural Language Toolkit or the Open Source Framework TensorFlow are used.

This enables the user to follow these steps in concrete terms using an easily accessible application and, in addition, to actively intervene in the code and thus better understand it.[1]

[1] Übersetzung durch www.deepl.com

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Projektarbeit bis auf die offizielle Betreuung selbst und ohne fremde Hilfe angefertigt habe und die benutzten Quellen und Hilfsmittel vollständig angegeben sind.

Soweit meine Rechte berührt sind, erkläre ich mich einverstanden, dass die Projektarbeit Angehörigen der Hochschule Emden/Leer für Studium / Lehre / Forschung uneingeschränkt zugänglich gemacht werden kann.

Osnabrück, den 31.01.2019 …….

(Christian Kitte)

Inhaltsverzeichnis

1 Einleitung

Künstliche Intelligenz und das maschinelle Lernen nehmen nicht nur gefühlt einen immer wichtiger werdenden Platz in der Gesellschaft ein. Während mit den konkreten Paradigmen der Umsetzung eher die in diesen Bereich Beschäftigten in Berührung kommen, sind die darauf basierenden Entwicklungen längst im Alltag angekommen.

Nicht nur in Form digitaler Assistenten wie Alexa von Amazon oder Cortana von Microsoft, sondern zumeist in Form von unsichtbaren Routinen werden sie in einer Vielzahl von Programmen und Anwendungen genutzt und vereinfachen oder verbessern die so erzielten Ergebnisse. Die so bereits stattfindende Entwicklung wird hierbei nicht langsamer, sondern eher stärker und durchdringender werden.

Ein wichtiger Teilaspekt ist das automatisierte Finden von Strukturen in scheinbar nicht strukturierten Daten und die darauf basierende Generierung von Wissen.

Im Rahmen dieser Arbeit wird eine Anwendung erstellt, welche die Abfrage des Dienstes Twitter anhand von Suchbegriffen oder dem Nutzernamen ermöglicht. Die so gewonnenen Daten werden in nachgelagerten Prozessen weiter vorbereitet, semantisch analysiert und abschließend visuell präsentiert. Hierzu wird zunächst auf die wichtigsten Grundlagen selbst eingegangen und im weiteren Verlauf ein Konzept für eine konkrete Anwendung sowie deren realen Umsetzung entwickelt.

Bei der Anwendung handelt es sich um ein durch Google modifiziertes und auf Google Drive ausgeführtes Python Notebook mit dem Namen Colaboratory oder kurz Colab. Diese Option bietet die Möglichkeit, mit nur geringem Aufwand konkret und begleitend den Weg vom Erhalt der Daten über deren Vorverarbeitung und semantischen Analyse bis hin zur Visualisierung zu verfolgen. Einzige Voraussetzungen hierfür sind ein Onlinezugang sowie ein Konto bei dem Dienst Twitter sowie Google Drive.

Durch den Anwendungstyp wird dem interessierten Laien gleichzeitig die Möglichkeit gegeben, sich auch interaktiv mit der beispielhaften Umsetzung auseinander zu setzen, diese zu erweitern, zu ergänzen oder aber zu verbessern. Ausdrücklich ist der Nutzer eingeladen, den vorliegenden Code als Basis für eigene Entwicklungen zu nutzen. Begleitet wird er hierbei durch eine ausführliche Dokumentierung.

Auf Grund der besseren Lesbarkeit wurden für geschützte Begriffe, Warennamen, Marken usw. keine Angaben zu den Rechten Dritter gemacht. Die Verwendung in dieser Arbeit berechtigt daher nicht zu der Annahme, dass diese frei von Rechten Dritter sind.

2 Grundlagen

2.1 Vom Data Mining zum Social Web Mining

Um das zugrunde liegende Projekt erfolgreich umzusetzen ist es zunächst notwendig, sich eingehender mit den Begrifflichkeiten des Data Mining und davon ausgehend dem Text und Social Web Mining zu beschäftigen.

Allen Ausprägungen gemein ist als Zielsetzung die Entdeckung von neuen Strukturen. Dies unterscheidet es vom Information Retrieval[2] (Informationsrückgewinnung[3]). Zielsetzung im Information Retrieval ist das Wiederauffinden von Informationen innerhalb großer und komplexer Datenbasen.

2.1.1 Data Mining

Jannaschk schreibt in (Jannaschk 2017, S. 1) über den Begriff Data Mining, dass es sich um ein Sammelbegriff für verschiedene Methoden und Techniken aus dem Bereich der Datenanalyse handle. Daher soll der Begriff an dieser Stelle etwas gründlicher betrachtet werden.

2.1.1.1 Vorüberlegungen zum Data Mining

Ziel des Data Mining ist es, aus einer gegebenen Menge von Daten vorhandene Strukturen zu erkennen und diese Erkenntnisse auf unbekannte Daten anzuwenden. Um diesen Prozess effizient zu gestalten, ist es notwendig, die für eine Fragestellung relevanten Daten zu identifizieren und zu extrahieren. Dies impliziert die Notwendigkeit, sich vor dem eigentlichen Prozess auf eine konkrete Fragestellung festzulegen.

Ein weiterer Punkt ist die Zuverlässigkeit und Reproduzierbarkeit der Datengewinnung. Hierzu schreibt Jannaschk: „Die Glaubwürdigkeit eines Analyseergebnisses hängt letztlich von der Nachvollziehbarkeit und der Systematik des gesamten Analyseprozesses ab. [...] Die Zuverlässigkeit eines Analyseergebnisses gibt darüber Auskunft, inwieweit es sich unter Verwendung analoger Verfahren reproduzieren lässt.“ (Jannaschk 2017, S. 3)

Weitergehend sieht er im Data Mining eine konkrete Methode zur Erkenntnisgewinnung und schreibt: „DM [gemeint ist Data Mining, Anm. d. Autors] ist als eine induktive Methode aufzufassen, bei der ein Anwender durch die Analyse von Daten zielgerichtet Informationen zu gewinnen sucht, die sich generalisieren bzw. mindestens auf andere unbekannte Daten übertragen lassen.“ (Jannaschk 2017, S. 8)

[2] Besonders im Zusammenhang von Texten wird i.d.R. der Begriff „Document Retrieval“ verwendet.
[3] „Information Retrieval“ wird manchmal auch etwas ungenau als Informationsbeschaffung übersetzt.

2.1.1.2 Maschinelles Lernen als Methode des Data Mining

Um Strukturen zu erkennen, bedarf es einen Algorithmus, der die vorliegenden und strukturierten Daten verarbeitet, analysiert, Merkmale extrahieren und gegebenenfalls nach einer Gesetzmäßigkeit sortieren kann. Erst hierdurch werden sie den Methoden des Data Mining zugänglich. Im einfachsten Fall sind dies programmierte Verzweigungen, welche jedoch schnell an Grenzen stoßen und nur bekannte Fälle verarbeiten können.

Die Lösung erreicht man durch Algorithmen, welche in der Lage sind, anhand von gelernten Beispielstrukturen ähnlicher Strukturen zu erkennen. Besser noch ist ein Algorithmus, der innerhalb von gegebenen Daten eigenständig neue Strukturen erkennen kann. Solche Algorithmen werden als lernende Algorithmen bezeichnet. Das Lernen selbst geschieht hierbei überwacht oder aber unüberwacht.

Bei den in der Praxis sehr erfolgreichen überwachten Lernen sind die zu erwartenden Ein- und Ausgabedaten bekannt, und können für die Lernphase verwendet werden. Kommen neuronale Netze zum Einsatz, so spricht man von Deep Learning. Bei einem unüberwachten Lernen wird versucht, innerhalb der vorliegenden Daten vorhandene Strukturen zu entdecken.

Im Kontext des Data Mining hat das maschinelle Lernen eine wichtige Rolle als Fähigkeit, im Rahmen von überwachtem Lernen gelerntes auf neue Daten anzuwenden und im Rahmen von unüberwachten Lernen, neue Zusammenhänge zu erkennen und anzuwenden. Somit ermöglicht es erst das maschinelle Lernen, Merkmale aus Daten zu extrahieren und weiter zu verwenden. Die Grenze zwischen maschinellem Lernen und anderen Methoden des Data Mining sind hierbei fließend.

2.1.1.3 Klassifikation, Segmentierung/Clustering und Regression

Die Beschäftigung mit den Aufgaben und Verfahren des Data Mining wird durch die Fülle an Begrifflichkeiten und Verfahren gerade zu Beginn sehr erschwert. Daher soll im folgendem auf drei wichtigen Begrifflichkeiten kurz eingegangen werden:

2.1.1.3.1 Klassifikation

Piazza sieht in (Piazza 2010, S. 42) die Aufgabe der Klassifizierung darin, „[…] Zusammenhänge in der Datenbasis zu ermitteln, die die Zugehörigkeit der Objekte zu einer Klasse beeinflussen und diese für die automatisierte Zuordnung der Objekte zu den Klassen zu verwenden.“ Sie gehören nach Piazza zu den überwachten Methoden.

Handelt es sich um eine binäre Klassifikation, so erfolgt sinngemäß nach (Müller und Guido 2017, S. 28) letztlich eine Aufteilung in eine von zwei möglichen Gruppen. Häufig werde in diesem Zusammenhang auch zwischen der positiven und negativen Klasse unterschieden, um den Bezug zur Fragestellung zu verdeutlichen.

2.1.1.3.2 Segmentierung und Clustering

Aufgabe der Segmentierung ist laut Piazza in (Piazza 2010, S. 45) die Gruppierung einer Menge möglichst ähnlicher („Interhomogenität“) Objekte in Gruppen („Cluster“). Die Mitglieder verschiedener Cluster sollen hierbei möglichst unähnlich („Intraheterogenität“) sein. Da die Gruppen im Vorfeld nicht bekannt seien, handele es sich bei Segmentierungsmethoden um Methoden des unüberwachten Lernens.

Nach Runkler gehört das Clustering ebenfalls zu den unüberwachten Lernverfahren, bei dem Daten Clustern zugeordnet werden. Die „Clustertendenz“ mache eine Aussage über das Vorhandensein von Clustern, das „Clustervaliditätsmaß“ hingegen quantifiziere die Güte eines Ergebnisses. (Runkler 2015, S. 109)

Das nachfolgende Bild zeigt beispielhaft, wie eine solche Segmentbildung aussehen könnte. Hierbei ist die Kreisform nicht zwangsläufig vorauszusetzen und kann verschiedensten Formen weichen.

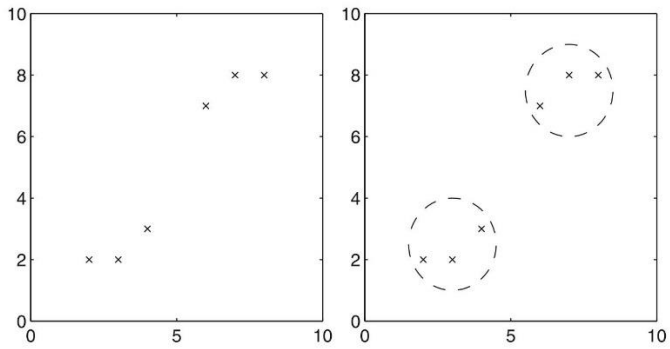


Abbildung 1: Beispielhafte Darstellung einer Clusterstruktur (Runkler 2015, S. 111)

2.1.1.3.3 Regression

Müller beschreibt als Ziel einer Regression die Vorhersage eines Wertes anhand der vorliegenden Daten und definierten Eigenschaften. Die Regression unterscheidet sich von der Klassifizierung dahingehend, dass bei der Ausgabe eine Kontinuität vorliegt, wohingegen das Ergebnis einer Klassifizierung ein entweder oder ist. (Müller und Guido 2017, S. 28)

Runkler führt dies in (Runkler 2015, S. 69) weiter aus. Die Umsetzung erfolge demnach mit Hilfe einer Regressionsanalyse. Diese schätze die Abhängigkeit zwischen Merkmalen, um darauf basierende Zusammenhänge zu erkennen und zu nutzen. Hierbei würden lineare Regressionsmodelle, die für Ausreißer weniger empfindliche robuste Regression sowie nichtlineare Regressionsmodelle unterschieden.

2.1.1.4 Ergebnisse des Data Mining

Ungeachtet des konkreten Verfahrens sind nach (Jannaschk 2017, S. 9) die Ergebnisse einer Analyse immer Strukturen innerhalb der untersuchten Daten. Diese Strukturen könnten beschrieben, deren Vorkommen an Eigenschaften festgemacht und die Wahrscheinlichkeit hierfür berechnet werden. Die gewonnene Information sei die Beschreibung dieser Strukturen und werde als „Muster“ bezeichnet und müsse vom Anwender validiert und eingeordnet werden.

Nimmt man das vorhergehend gesagte, so lässt sich mit einem gefundenen Muster zunächst nicht viel anfangen. Erst durch die Integration in ein geeignetes Modell kann es zu Vorhersagen genutzt werden. Nur innerhalb einer Modellwelt führt die gewonnene Information zu einer Zustandsänderung innerhalb des Modells und kann somit Rückschlüsse auf die reale Welt geben.

2.1.2 Text Mining

Text Mining unterscheidet sich vom Data Mining in der Art der untersuchten Daten. Stehen beim Data Mining aufgezeichnete, berechnete oder gemessene numerische Daten zur Verfügung, die direkt mit Hilfe eines geeigneten Algorithmus untersucht werden können, so sind es beim Text Mining Texte. Diese müssen vor der weiteren Verwendung einer semantischen Analyse unterzogen werden.

Grundsätzlich gelten die zuvor gemachten Aussagen zum Data Mining auch für das Text Mining. Auch hier ist das Mining nur eine Etappe in einem größeren Prozess und bedarf der sorgfältigen Vorplanung.

2.1.2.1 Arten von Texten

Bei der Untersuchung von Texten können nach (Müller und Guido 2017, S. 307ff.) vier unterschiedliche Arten von Texten unterschieden werden:

1. „Kategorische Daten“, die einer definierten Listen entsprängen und so einfach in Gruppen eingeteilt werden könnten wie z.B. Farbangaben,
2. „Freie Strings“, welche sich „semantischen Kategorien“ zuordnen ließen,
3. „Strukturierte Stringdaten“, denen eine Struktur innewohne wie Adressdaten, Namen, Ortsangaben etc. sowie
4. „Textdaten“, denen als freie Texte keine Struktur unterläge.

Um Texte dem Data Mining zugänglich zu machen, ist es notwendig, die Informationen des Textes zu extrahieren. Diese manifestieren sich in den genutzten Wörtern und dem Aufbau des Satzes. Ziel jeder Aufbereitung ist es also, einen Text bewertbar zu machen.

Im Zusammenhang mit der Textanalyse wird nach (Müller und Guido 2017, S. 309) die zugrundeliegende Datenbasis als „Korpus“, ein Datensatz oder Datenpunkt hieraus als „Dokument“ bezeichnet.

2.1.2.2 Bag-of-Word

Eine einfache und häufig eingesetzte Möglichkeit, Texte dem Data Mining zugänglich zu machen, ist die Bag-of-Words Methode. „Mit dieser Repräsentation ignorieren wir einen Großteil der Struktur des Eingabetextes wie Kapitel, Absätze, Sätze und Formatierung und zählen lediglich, *wie häufig jedes Wort in jedem Text im Korpus vorkommt* [sic].“ (Müller und Guido 2017, S. 311f.)

In der folgenden Abbildung sind die einzelnen Schritte des Verfahrens, ausgehend vom ursprünglichen Satz oben bis hin zum Wordvektor unten zu sehen:

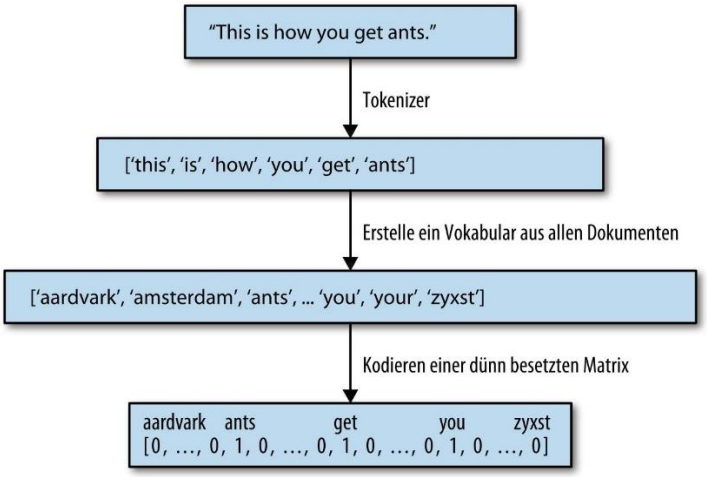


Abbildung 2: Verarbeitungsschritte eines Bag-of-Word Ansatzes (Müller und Guido 2017, S. 312)

1) Im Vorfeld wird ein Korpus mit für die Auswertung dienlichen Dokumenten erstellt.
2) Aus den im Korpus vorkommenden Wörtern (Token, Term) wird ein Vokabular erstellt. Diesen Vorgang bezeichnet man als Tokenisierung. Die Anzahl der im Vokabular vorhandenen Token definiert die Dimension des im nächsten Schritt genutzten, dünn besetzten Vektors. Jede Stelle dieses Vektors steht hierbei für ein definiertes Token.
3) Indem anhand der im Dokument enthaltenen Token in den zuvor erhaltenen Vektor die diesen entsprechenden Stellen mit 1 gesetzt werden, erhält man Dokument- bzw. Wortvektoren. Alle Dokumentvektoren untereinander ergeben die Dokumentmatrix. Hierbei handelt es sich um eine dünn besetzte Matrix.

Um mit den so aufbereiteten Daten arbeiten zu können, müssen im Rahmen eines Scorings die Token bewertet werden. Bereits durch die Bildung der Dokumenten-Vektoren erhält man

ein einfaches, binäres Scoring. Durch eine zusätzliche Bewertung z.B. in eine eher positive oder negative Bedeutung der Token erhält der Vektor selbst eine Bedeutung.

In Variationen dieses Vorgehens werden nach (Müller und Guido 2017, S. 323) nicht nur einzelne Token (Unigramme), sondern auch Wortgruppen von zwei und mehr Token berücksichtigt. Die erhaltenen Ergebnisse heißen Bigramme (zwei Token), Triplette bzw. Trigramme (drei Token) oder allgemein n-Gramme (Folge von n Token).

In der Praxis wird diese Aufgabe im Rahmen einer Anlernphase durchgeführt. Dem System werden hierbei bewertete Texte vorgelegt, mit deren Hilfe ein überwachtes Lernen stattfindet. Hierzu werden die Texte vektorisiert und anschließend bewertet. Die hierfür notwendigen Trainingsdaten können durch manuelle Sichtung von Texten und deren Klassifizierung gewonnen werden. Als Alternative bieten sich vorbereitete Trainingsdaten an.

Als Ergebnis liegen klassifizierte Texte in einer dem maschinellen Lernen zugänglichen Form vor. Hierbei ist es wichtig zu verstehen, dass das Ziel von Methoden wie der hier vorgestellten Bag-of-Word Methode die Verfügbarmachung von Texten für die Methoden des maschinellen Lernens und des Data Mining ist.

Um die Qualität der Daten zu verbessern, können weitere Maßnahmen ergriffen werden. So kann zusätzlich die Häufigkeit des Vorkommens einzelner Terme für deren Gewichtung herangezogen werden. Seltenere Wörter werden so stärker gewichtet. Ein verbreitetes Verfahren hierfür ist das term frequency – inverse document frequency (tf-idf) Verfahren.

2.1.2.3 Term Frequency – Inverse Document Frequency

Die folgenden Ausführungen beziehen sich in Teilen auf den Ausführungen in (Wikipedia-Autoren 03.10.2018). Hiernach gibt die Häufigkeit eines Begriffs (term frequency, tf) das absolute Vorkommen eines Terms $\#(t, D)$ in einem Dokument an. Um Verzerrungen zu vermeiden, kann diese Angabe normalisiert werden. Hierdurch erhält man die relative Häufigkeit $tf(t, D)$. Hierfür wird die absolute Häufigkeit durch die maximale Häufigkeit $max_{t' \epsilon D} \#(t', D)$ des Vorkommens geteilt:

$$\mathrm{tf}(t, D) = \frac{\#(t, D)}{\max_{t' \in D} \#(t', D)}$$

Formel 1: Relative Vorkommenshäufigkeit eines Terms (Wikipedia-Autoren 03.10.2018)

Die inverse Dokumenthäufigkeit (inverse document frequency, idf) ist hingegen ein Maß über das Vorkommen eines Terms in allen Dokumenten des Korpus. Hierfür wird der Logarithmus

aus der Gesamtzahl der Dokumente N durch die Anzahl der Dokumente, welche den Term enthalten, gebildet:

$$\mathrm{idf}(t) = \log \frac{N}{\sum_{D:t \in D} 1}$$

Formel 2: Inverse Dokumentenhäufigkeit eines Terms (Wikipedia-Autoren 03.10.2018)

Durch die Kombination beider Angaben kann ein Maß für die Gewichtung tf-idf eines Terms ermittelt werden. Je weniger Dokumente einen Term erhalten, umso stärker ist dessen Informationsgehalt:

$$\mathrm{tf.idf}(t, D) = \mathrm{tf}(t, D) \cdot \mathrm{idf}(t)$$

Formel 3: Gewicht eines Terms nach tf-idf (Wikipedia-Autoren 03.10.2018)

2.1.2.4 Weitere Optimierungen

Eine weitere Möglichkeit, die Qualität zu verbessern findet bereits im Vorfeld und auf Basis der verwendeten Texte mit dem Ziel statt, Wörter ohne Informationsgehalt zu eliminieren, solche mit ähnlichem Informationsgehalt zusammen zu fassen. Beispiele hierfür sind:

- Vereinheitlichung der Schreibung
- Vereinheitlichung der Sprache
- Verwendung der Grundform (Lemma)
- Weglassen häufiger Füllworte (Stopworte)

2.1.3 Social Web Mining

Eine besondere Form des Text Minings ist das Social Web Mining. Hierbei basieren die zu untersuchenden Texte auf den Nachrichten sozialer Plattformen wie Twitter oder Facebook.

Zusätzlich zu den enthaltenen Texten können Metainformationen wie beispielsweise die Herkunft, das Datum der Erzeugung oder Weiterleitungen als zusätzliche Information genutzt werden.

2.2 TensorFlow

TensorFlow wird in (Hope et al. 2018, S. 2) als „[…] Googles System zweiter Generation zum Implementieren und Betreiben von Deep-Learning-Netzen […]“ beschrieben. Weiter führen die Autoren aus, dass es im November 2015 als Open-Source-Framework unter der Apache-Lizenz 2.0 veröffentlicht wurde.

Der Name TensorFlow ist ein Kunstwort und leitet sich aus den Wörtern Tensor und Flow ab. In (Hope et al. 2018, S. 5) werden die Tensoren folgendermaßen beschrieben: „Tensoren sind die im Deep Learning übliche Erscheinungsform von Daten. Einfach ausgedrückt sind Tensoren nichts weiter als mehrdimensionale Felder, eine Erweiterung zweidimensionaler Tabellen (Matrizen) auf Daten mit mehr Dimensionen.“ Flow steht im Englischen für fließend und wird hier sinnbildlich für das Fließen von Daten verwendet.

Bei Tensor Flow handelt es sich somit einfach gesagt um ein Framework, welches Daten bearbeitet und diese bearbeiteten Daten anschließend weiterleitet. Die Darstellung der Verarbeitungsschritte erfolgt hierbei nach Hoppe als „Datenflussgraph“. Die nachfolgende Darstellung verdeutlicht dies:

Links sind die in einer zweidimensionalen Matrix angeordneten Daten dargestellt und symbolisieren den Tensor. Auf der rechten Seite sind die auf den Daten auszuführenden Berechnungen in Form eines Graphen aufgeführt. Nach jedem Schritt fließen die Daten bildlich gesprochen zum nächsten Schritt der Bearbeitung.

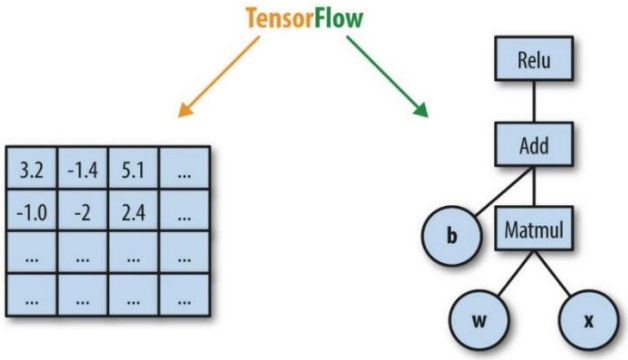


Abbildung 3: Datenfluß-Berechnungsgraph nach Hope (Hope et al. 2018, S. 5)

2.2.1 Verfügbarkeit

TensorFlow ist in der Programmiersprache C++ geschrieben. Es ist für den verteilten Einsatz auf unterschiedlichen Betriebssystemen konzipiert und unterstützt über verschiedene Versionen neben Windows auch Linux, macOS sowie Raspberry[4]. Daneben ist Tensor Flow als Docker Container[5] sowie als Jupyter Notebook[6] verfügbar. Auf Jupyter Notebooks wird im Rahmen dieser Arbeit im Abschnitt 2.3.2.3 - *IPython und Jupyter* kurz eingegangen. Gerade das Notebook bietet sich für einen Einstieg an, da es ohne eine Installation sofort eingesetzt werden kann.

[4] Eine ausführliche Anleitung zur Installation unter den beschriebenen Betriebssystemen findet man auf der Website des Projekts unter https://www.tensorflow.org/install/.
[5] Unter https://hub.docker.com/r/tensorflow/tensorflow/ finden sich weitere Informationen.
[6] Unter https://colab.research.google.com/notebooks/welcome.ipynb finden sich weitere Informationen.

Somit stellt TensorFlow eine Infrastruktur für das maschinelle Lernen sowie der Konzeption, dem Training und Betrieb neuronaler Netze zur Verfügung, die auch unabhängig auf Basis eines Desktops eingesetzt werden kann. Hierdurch werden der Zugang und Einstieg in diesen Bereich erheblich vereinfacht.

2.3 Python

Python schaffte es in den letzten Jahren, sich vor allem im Bereich der KI und des maschinellen Lernens neben der von Microsoft ins Leben gerufenen Sprache R einen Namen zu machen. Übersehen wird hierbei, dass diese Sprache daneben für einen wesentlich breiteren Bereich einsetzbar ist.

Die Sprache wurde zu Anfang der 1990 er Jahre ins Leben gerufen. Die aktuell immer noch gültige Hauptversion 3.0 wurde im Jahr 2008 veröffentlicht (vergl. Steyer 2018, S. 5).

An dieser Stelle soll und kann keine tiefergehende Einführung in Python oder der Programmierung erfolgen. Für Leser, die sich tiefer mit dieser Sprache beschäftigen möchten, finden sich im Internet eine Vielzahl an Quellen[7].

2.3.1 Einordnung der Sprache

Python ist eine interpretierte Sprache, deren fertiger Code erst während der Ausführung von einem Interpreter übersetzt. Laut (Wikipedia-Autoren 24.09.2018) ist „CPython oder cPython […] die in der Programmiersprache C geschriebene Referenzimplementation des Python Interpreters. Er wird auch oft nur Python genannt." Daneben existieren weitere Interpreter wie PyPy[8] oder IronPy[9].

Python ist nicht auf ein einzelnes Betriebssystem festgelegt. Es lässt sich auf Windowssystemen einfach installieren und ist standardmäßig in den meisten Linuxversionen vorinstalliert. Laut (Autoren ApfelWiki 29.03.2009) wurde Python seit MacOSX 10.2 von Apple mit ausgeliefert und kann somit auf diesen Macs auch ausgeführt werden. Ebenso gibt es eine Reihe freier und kommerzieller IDEs für diese Sprache.

Die Sprache wurde von Grund auf objektorientiert aufgebaut, ist jedoch als eine Multiparadigmensprache konzipiert. So schreibt Steyer in (Steyer 2018, S. 4): „Python unterstützt sowohl die objektorientierte, die aspektorientierte, die strukturierte als auch die funktionale Programmierung." Zudem sei ein „[…] zentrales Feature […] in Python die dynamische Typisierung samt dynamischer Speicherbereinigung. Damit kann man Python auch als reine Skriptsprache nutzen." (Steyer 2018, S. 4)

---

[7] Unter https://www.python-kurs.eu/index.php und https://www.python.org/doc/ finden sich geeignete Tutorien.
[8] Unter https://pypy.org/ ist die Website des Projektes zu finden.
[9] Unter http://ironpython.net/ ist die Website des Projektes zu finden.

2.3.2 Charakteristik und Vorteile

An dieser Stelle wird im Folgenden kurz auf einige charakteristische Eigenschaften und Ausprägungen der Sprache eingegangen. Ziel ist hierbei, dem Leser ein Gefühl für die Sprache zu geben.

2.3.2.1 Klarheit und Einfachheit

Steyer schreibt hierzu sinngemäß in (Steyer 2018, S. 3), dass bei der Entwicklung von Python einer der Schwerpunkt die gute Lesbarkeit des Quellcodes gewesen sei. Python nutze weniger Schlüsselwörter als andere Sprachen und verwende weniger syntaktische Konstruktionen. Zudem nutze es zur Strukturierung statt Klammer oder Semikolon die Einrückung. Der so entstehende Code sei deutlich übersichtlicher, kürzer und weniger anfällig für Fehler.

Nach Steyer war dies somit eine bewusste Entscheidung und präferiert die Logik der Anwendung vor dem Prozess der Optimierung.

2.3.2.2 Erweiterungsfähigkeit

Python selbst verfügt bereits standartmäßig über eine große Anzahl an Bibliotheken, der Python Standard Library[10]. Zu dieser kommt eine sehr große Anzahl an Paketen aus verschiedensten, oft ebenfalls aus dem Open Source Bereich stammenden, Projekten hinzu. Das Themengebiet ist hierbei weit gefächert und erfasst aktuell[11] 153.183 verzeichnete Projekte.

Pakete können hierbei auf einfachem Wege mit Hilfe des Python eigenen Paketmanagers pip (ein rekursives Akronym für pip installs packages) installiert werden. Eine besondere Stärke von Python ist hierbei, dass Module auch in einer anderen Sprache wie beispielsweise C geschrieben werden können. Hierbei gestaltet sich die Erstellung eigener Erweiterungen als eher einfach und gut dokumentiert[12].

Gleichzeitig lassen nach Steyer mit Hilfe von Python Module und Plug-ins für andere Programme wie LibreOffice, SPSS, GIMP oder Blender umsetzen. (Steyer 2018, S. 3)

2.3.2.3 IPython und Jupyter

Häufig stößt man besonders im wissenschaftlichen Umfeld von Python auf die Begriffe Python Notebook, IPython Notebook oder aber Jupyter Notebook.

Das Akronym IPython steht für Interactive Python und spielt auf die Fähigkeit von Python an, nicht nur fertige Programme oder Skripte ausführen, sondern auch interaktiv genutzt werden zu können. Die Fortführung dieses Konzeptes führt zum Python Notebook.

---

[10] Unter https://docs.python.org/3/library/index.html findet sich ein Verzeichnis aller registrierten Module.
[11] Im Oktober 2018.
[12] Unter https://docs.python.org/3/extending/index.html finden sich hierzu weitere Informationen.

Auf der Projektseite unter (IPython 23.07.2018) wird die Aufgabe eines Notebooks wie folgt beschrieben: “The notebook extends the console-based approach to interactive computing in a qualitatively new direction, providing a web-based application suitable for capturing the whole computation process: developing, documenting, and executing code, as well as communicating the results.”[13]

Es erweitert hiernach die interaktiven Fähigkeiten und stellt eine Webanwendung für die Interaktion zur Verfügung. Entwicklung, Dokumentation und Ausführung verschmelzen, die Kommunikation von Lösungen wird vereinfacht. Eine Webanwendung wird dann auch beschrieben als: “[…] a browser-based tool for interactive authoring of documents which combine explanatory text, mathematics, computations and their rich media output.”[14] (IPython 23.07.2018)

Die bei der Arbeit entstehenden Dokumente beinhalten alle Bestandteile, um fertige Lösungen nachvollziehen zu können. Hierbei handelt es sich um einfach weitergebbare Dateien. Dem entsprechend werden sie auf der Projektseite beschrieben als “[…] a representation of all content visible in the web application, including inputs and outputs of the computations, explanatory text, mathematics, images, and rich media representations of objects.”[15] (IPython 23.07.2018)

Im Jahr 2014 teilte sich das Projekt in die Zweige IPython, welches die Weiterentwicklung eines interaktiven Pythons zum Ziel hat und Jupyter. Der Schwerpunkt von Jupyter liegt in der zukünftigen Weiterentwicklung der Sprachunabhängigen Komponenten wie dem Notebook Format, der Webapplikation oder des Nachrichtenprotokolls.

2.4 Twitter

Twitter ist ein im April 2006 (siehe Twitter inc. 2018) gegründeter Onlinedienst mit Hauptsitz in San Francisco, der es Nutzern ermöglicht, in einer standardisierten Form Nachrichten bestimmter Länge und Form online auf deren Plattform zu veröffentlichen. Hierfür ist vom Nutzer ein Benutzerkonto anzulegen. Der Zugang zu dem Dienst selbst kann im Anschluss über eine Vielzahl mobiler Endgeräte oder mit jedem internetfähigen Endgerät direkt über ein Webportal erfolgen. Hierbei unterstützt Twitter nach eigenen Angaben über 40 Sprachen (vergl. Twitter inc. 2018).

---

[13] Das Notebook erweitert den konsolenbasierten Ansatz auf interaktives Computing in eine qualitativ neue Richtung und bietet eine webbasierte Anwendung, die geeignet ist, den gesamten Berechnungsprozess zu erfassen: Entwicklung, Dokumentation und Ausführung von Code sowie die Kommunikation der Ergebnisse. (Übersetzung durch www.deepl.com)

[14] …ein browserbasiertes Werkzeug zur interaktiven Erstellung von Dokumenten, die erklärenden Text, Mathematik, Berechnungen und deren Rich-Media-Ausgabe kombinieren. (Übersetzung durch www.deepl.com)

[15] …eine Darstellung aller in der Webanwendung sichtbaren Inhalte, einschließlich der Ein- und Ausgaben der Berechnungen, des erklärenden Textes, der Mathematik, der Bilder und der Rich-Media-Darstellungen von Objekten. (Übersetzung durch www.deepl.com)

Laut dem Mitteilungsblatt für Investoren für das zweite Quartal 2018 in (Twitter inc. 2018) verfügt der Dienst aktuell über rund 3500 Mitarbeiter in 35 Niederlassungen weltweit. Die Anzahl der monatlich aktiven Nutzer wird mit rund 335 Millionen angegeben, wovon 68 Millionen aus den USA stammen. Im zweiten Quartal 2018 habe der Zuwachs hierbei bei 11% gelegen.

2.4.1 Kommunikationskonzept

Das hinter Twitter stehende Konzept ist grundsätzlich sehr einfach und soll hier in Kürze dargestellt werden[16]:

- Eine Nachricht wird als „Tweet“ bezeichnet und hat eine maximale Länge von 140 Zeichen. Verweise und Anhänge werden nicht mitgezählt.
- Ein einem vorgestellten „@“ (At-Zeichen) wird ein anderer Nutzer als Empfänger angegeben.
- Mit einem vorgestellten „#“ (Hashtag) kann eine Nachricht einem neuen oder vorhandenen Thema zugeordnet werden.
- Einem Nutzer können andere User als „Follower“ folgen und erhalten so alle von ihm erstellten Tweets.
- Analog hierzu kann ein User als „Follower“ anderen Usern oder Themen folgen. Er erhält in diesem Fall alle von den Usern oder zu den Themen generierten Tweets.
- Ein empfangener Tweet kann vom Empfänger unverändert („Retweet“) oder ergänzt („Quote Tweet“) weitergeleitet („retweeted“) werden.

2.4.2 Twitter als Datenbasis

Twitter schreibt über sich selbst: „Twitter is what’s happening in the world and what people are talking about right now.”[17] (Twitter inc. 2018)

Geht man allein von der im Anfang dieses Kapitels genannten Zahl an monatlich aktiven Nutzern aus, so verdeutlicht dies die Menge der in Twitter potentiell entstehenden Daten, selbst unter der Annahme, dass nicht jeder Nutzer täglich aktiv Tweets verfasst. Durch das Konzept der Follower kommt es zu einer Multiplikation.

Zu hinterfragen ist jedoch die Qualität der so entstehenden Daten. Auf die Qualität und Objektivität von Twitter als Datenquelle einzugehen ist jedoch nicht Inhalt dieser Ausarbeitung und würde deren Rahmen bei weitem überschreiten.

[16] Eine ausführlichere Darstellung findet sich im Internet unter https://help.twitter.com/de.
[17] Twitter ist das, was in der Welt passiert und worüber die Leute gerade reden. (Übersetzung durch www.deepl.com)

3 Zielsetzung und grundlegende Anforderungen

Nachfolgend soll kurz auf die grundlegenden Anforderungen und Ziele des praktischen Teils dieser Arbeit eingegangen werden. Im Fokus stehen hierbei die folgenden Aspekte:

- Grundsätzliche Zielsetzung
- Zielgruppe der Anwendung
- Umsetzung als Python Notebook
- Anforderungen an die Anwendung

3.1 Grundsätzliche Zielsetzung

Die zu erstellende Anwendung soll es einem Nutzer ermöglichen, auf eine einfache und intuitive Art einen Einblick in die Thematik maschinelles Lernen und Data Mining zu erhalten. Die Anforderungen an die eingesetzte Hard und Software sollen hierbei möglichst gering sein. Dies gilt entsprechend für das vorausgesetzte Vorwissen.

Dem Nutzer soll hierzu ein beispielhafter, vollständiger Workflow vorgestellt werden, der in allen Aspekten nachzuvollziehen und änderbar ist. Hierzu gehört die begleitende Dokumentation durch die verschiedenen Stufen der Anwendung.

3.2 Zielgruppe der Anwendung

Der Zielgruppe der hier zu entwickelnden Anwendung gehören tendenziell eher Personen ohne nähere Erfahrung in diesem Bereich an.

Mitglieder der Zielgruppe im Sinne dieser Anwendung

- …können mit einem Desktoprechner und ihrem gewohnten Betriebssystem souverän umgehen. Einfache Arbeiten und Installationen stellen für sie kein Problem dar.
- …können sich im Internet souverän bewegen. Downloads oder die Suche nach Informationen stellen für sie kein Problem dar.
- …Können grundsätzlich Begriffe wie maschinelles Lernen oder künstliche Intelligenz einordnen, ohne über ein tieferes Wissen darüber zu verfügen.
- …verfügen idealerweise über grundlegende Kenntnisse in Python, so dass einfacher Code nachvollzogen werden kann.

Die Anforderungen an die Zielgruppen werden in der folgenden Abbildung nochmals zur Verdeutlichung wiedergegeben:

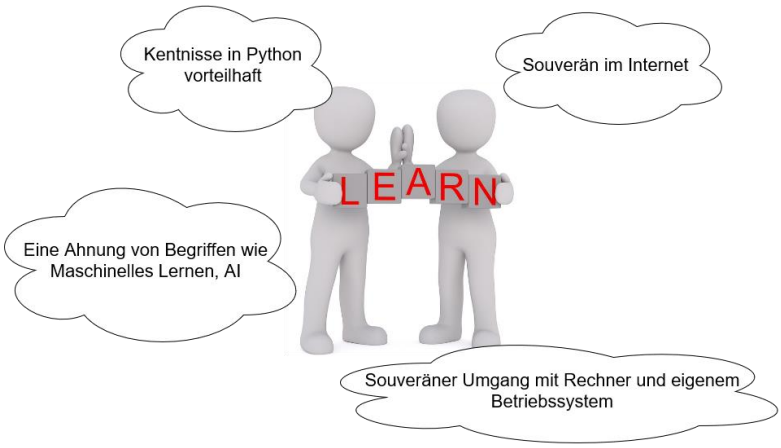


Abbildung 4: Beschreibung der verwendeten Zielgruppe

3.3 Umsetzung als Python Notebook

Die Umsetzung der Anwendung erfolgt auf Basis eines Python Notebooks (siehe hierzu auch 2.3.2.3 - *IPython und Jupyter*).

Durch die browserbasierte Ausführung kann ein Notebook grundsätzlich unter allen Betriebssystemen lokal ausgeführt und bearbeitet, jedoch auch online gehostet werden. In diesem Falle sind für den Zugriff nur eine Internetverbindung sowie ein aktueller Browser notwendig. Hierfür bieten sich etablierten Plattformen wie Google Drive oder GitHub an.

Um die Anforderungen und Hürden an die Nutzung dieser Anwendung gering zu halten, wird die Umsetzung mit Hilfe eines online gehosteten Notebooks erfolgen. Als Plattform bietet sich wegen der geringen Hürden bei der Erstellung eines Accounts Google Drive an.

3.4 Anforderungen an die Anwendung

Aus dem Titel dieser Arbeit und den zuvor gemachten Ausführungen ergeben sich die konkreten Anforderungen nach der hier zu erstellenden Anwendung.

Als Ergebnis des praktischen Teils dieser Arbeit wird eine funktionstüchtige Anwendung in Form eines online gehosteten Python Notebooks verfügbar sein, welches exemplarisch das notwendige Vorgehen aufzeigt, um mittels definierter Suchbegriffe oder Nutzernamen Tweets aus Twitter zu selektieren, den so entstehenden Korpus im Folgenden für eine weitere Verarbeitung vorzubereiten, zu klassifizieren und abschließend zu visualisieren.

Im Zentrum steht hierbei die semantische Analyse der Textnachricht zur Klassifizierung in eher negative, neutrale sowie positive Tweets. Die Umsetzung erfolgt unter Zuhilfenahme des Frameworks TensorFlow sowie vortrainierter Modelle. Zusätzlich sollen Auswertungen zum Wortgebrauch sowie der Länge der Tweets in Bezug zu Ihrer Semantik sowie der Anzahl an Likes und Retweets durchgeführt werden.

Als Sprache wird Python in Verbindung mit gängigen Modulen und Bibliotheken eingesetzt werden, wie sie auch in der Praxis weit verbreitet und im Einsatz sind.

Jeder im Rahmen dieser Arbeit ausgeführte Bearbeitungsschritt soll einfach, transparent und direkt nachvollziehbar sein. Hürden in Form von notwendigen Installationen sollen auf ein Minimum reduziert werden. Daher wird eine online gehostete Variante eines Python Notebooks bevorzugt. Ausdrücklich soll die Anwendung nicht für einen produktiven Einsatz konzipiert sein, sondern für die Lehre.

Auch ist es gerade erwünscht und gewollt, dass der Nutzer direkt in der Lage ist, selbst aktiv in den Code einzugreifen und so den Ablauf der Anwendung zu ändern. Am Ende soll der Leser in der Lage sein, ein ähnlich gelagertes Problem selbst softwaretechnisch zu lösen und sich weiter in die entsprechende Materie einarbeiten zu können.

4 Konzeptionelle Umsetzung

Nachdem im vorhergehenden Kapitel die grundsätzliche Zielsetzung und Anforderungen diskutiert wurden, soll im Folgenden auf die konzeptionelle Umsetzung des praktischen Teils dieser Arbeit eingegangen werden. Anhand der hierbei erarbeiteten Vorgehensweise folgt im Anschluss die konkrete Umsetzung.

4.1 Twitter als Datenquelle

Im Rahmen dieser Arbeit wird Twitter als Datenquelle der von uns genutzten Texte dienen. Auf den folgenden Kapiteln soll das hierzu notwendige Vorgehen erörtert werden. An dieser Stelle muss erwähnt werden, dass neben der Möglichkeit, selbst Daten zu sammeln grundsätzlich auf darauf spezialisierte, kommerzielle Dienste zurückgegriffen werden kann. Das eingehen hierauf würde jedoch den Rahmen dieser Arbeit überschreiten.

Im Folgenden stehen die folgenden vier Punkte im Fokus:

- Voraussetzung für den Zugriff
- Möglichkeiten des Zugriffs
- Vorgehen bei der Suche
- Rohform und Speicherung der Daten

4.1.1 Voraussetzungen für den Zugriff

Der gezielte Zugriff auf die Daten des Dienstes ist ohne die Einrichtung eines Accounts nicht möglich. In diesem Fall hat man Zugriff auf die Tweets von Nutzern oder Themen und kann selbst aktiv tätig werden. Sollen jedoch Daten von Twitter für eine spätere Auswertung gesammelt werden, so reichen die so erhaltenen Datenmengen in der Regel nicht aus.

Für eine umfassende Auswertung ideal wäre ein Zugriff auf den gesamten Datenbestand, den Twitter jedoch nur einem sehr kleinen Kreis ausgewählter Forschungsprojekte und einigen Partnerunternehmen und Instituten in den USA wie dem MIT (Massachusetts Institute of Technology) oder IBM (International Business Machines) einräumt. (vergl. Pfaffenberger 2016, S. 43)

Um eine für die Auswertung geeignete Menge an Daten zu erhalten, muss neben dem normalen Twitteraccount ein zusätzlicher Entwickler-Account angelegt werden, der es ermöglicht, in einem größeren Umfang auf die von Twitter angebotene API (Application Programming Interface) zugreifen zu können.

Hierfür wird ein mehrseitiger Anmeldeprozess durchlaufen, in dessen Rahmen auch Angaben über die beabsichtigte Nutzung zu machen sind. Erst im Anschluss daran kann eine konkrete Anwendung (App) angemeldet werden. Hierbei unterscheidet Twitter zwischen nutzer- (User

Auth) oder aber anwendungsbezogen (App Auth) Anwendungen. Einen Überblick über die einzelnen Schritte der Anmeldung findet sich im Anhang in Kapitel (1).

In der nachfolgenden Abbildung wird dies nochmals verdeutlicht. Ein zugreifendes Programm muss innerhalb eines Entwickler-Accounts registriert sein. Dieses bedingt einen normalen Twitter-Account. Erst nach der Registrierung ist ein Zugriff auf die Twitter API möglich. Hierfür werden die bei der Registrierung erhaltenen Token genutzt. Diesen Zusammenhang verdeutlicht das nachfolgende Diagramm.

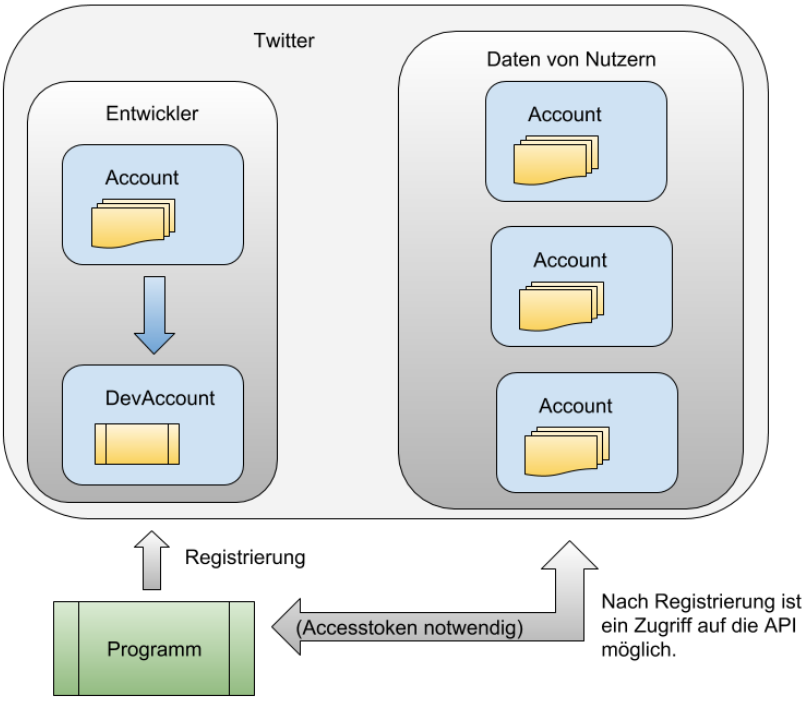


Abbildung 5: Zusammenhang zwischen Entwickler-Account, Anwendung und Nutzerdaten

Über die Nutzung der Anwendungsdaten schreibt Pfaffenberger in (Pfaffenberger 2016, S. 43), dass für den Zugriff der Counsumer Key und das Consumer Secret verlangt werden. Erfolge ein paralleler Zugriff über dieselbe App, so müsse ein Access Token bzw. ein Access Token Secret übergeben werden.

Für die Autorisierung kommt OAuth (Open Authorization) zum Einsatz. Wikipedia sagt hierzu in (Wikipedia-Autoren 20.09.2018):

> „**OAuth** (Open Authorization) ist ein offenes Protokoll, das eine standardisierte, sichere API-Autorisierung für Desktop-, Web- und Mobile-Anwendungen erlaubt. Es wurde ab 2006 entwickelt und 2007 in der ersten Version veröffentlicht.

> Ein Endbenutzer (*User* oder *Resource Owner*) kann mit Hilfe dieses Protokolls einer Anwendung (*Client* oder *Third-Party*) den Zugriff auf seine Daten erlauben (*Autorisierung*), die von einem anderen Dienst (*Resource Server*) bereitgestellt werden, ohne geheime Details seiner Zugangsberechtigung (*Authentifizierung*) dem Client preiszugeben. Der Endbenutzer kann so Dritte damit beauftragen in seinem Namen einen Dienst zu konsumieren. Typischerweise wird dabei die Übermittlung von Passwörtern an Dritte vermieden."

Auch bei der Verwendung gemäß der oben gezeigten Abbildung kommt es zu keiner direkten Aktion der einzelnen Nutzer und der Anwendung. Vielmehr reguliert Twitter den Zugriff anhand des tokenbasierten Verfahrens.

4.1.2 Möglichkeiten des Zugriffs

Grundsätzlich erfolgt jeder Zugriff auf Twitter über die einer Anwendung verfügbar gemachten API. Diese umfasst neben der Suche nach und dem Erhalt von Tweets auch diverse andere Möglichkeiten wie den Versand von Direktnachrichten (Direct Massage API), den Zugriff auf Accountdaten (Account Activity API) oder die Unterstützung von Kampagnen (Ads API).

Die Möglichkeiten und Skalierbarkeit des Zugriffs selbst richten sich stark am gewählten Account aus. So sind die kostenlosen Accounts wesentlich restriktiver ausgerichtet als kommerzielle und kostenpflichtige. Dies betrifft zum einen den Umfang der Möglichkeiten als auch die Menge der Zugriffe. Hierzu werden maximale Zugriffe innerhalb eines definierten Zeitfensters, beispielsweise 1000 Aufrufe je 15 Minuten, definiert.

Für das Sammeln von Tweets kann zum einen auf die Streaming API zurückgegriffen werden, mit deren Hilfe Tweets Live gestreamt werden können. Die zweite Möglichkeit ermöglicht es, mit Hilfe der Search API Daten aus der Vergangenheit zu erhalten. Hierfür kann über die von Twitter angebotene API eine Anfrage erstellt werden. Teil der Anfrage sind Kriterien zu Abfrage sowie die Menge der gewünschten Tweets. Alternativ kann auch eine Nutzername als Kriterium genutzt werden. Auf die letztere wird im Rahmen dieser Anwendung zugegriffen werden.

Den zuvor genannten Zusammenhang gibt auch die folgende Übersicht wieder. Ein Programm, hier das im Rahmen dieser Arbeit eingesetzte Modul Tweepy im blauen Kasten, nutzt über das Internet verfügbare und in grün dargestellte API-Schnittstellen von Twitter und erhält Feedback in Form von Daten (hier gelb dargestellt).

Die erhaltenden Daten (hier orange dargestellt) sind unter anderen als im JSON-Format formatierte Strings verfügbar und lassen sich somit einfach speichern.

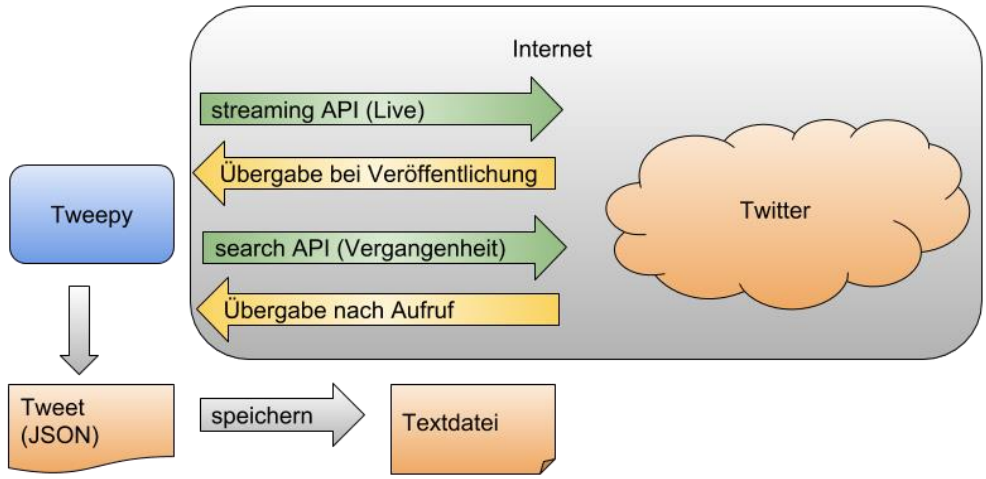


Abbildung 6: Übersicht über den Abruf von Daten des Dienstes Twitter, hier durch die grün dargestellten API Aufrufe von und der gelben Rückgabe von Daten an Tweepy dargestellt. Die Rohdaten werden im JSON-Format zurückgegeben. Über Tweepy sind sie objektorientiert oder im JSON-Format verfügbar.

4.1.3 Vorgehen bei der Suche

Grundsätzlich macht Twitter selbst keine Vorgaben bezüglich der gestellten Abfragen, bietet aber auch keine Hilfe an. Um die für eine Beantwortung geeigneten Daten zu erhalten, spielt die Auswahl geeigneter Filter[18] eine wichtige Rolle. Ein Filter definiert beispielsweise die Stichwörter, die Zeit oder die Lokalisierung der gesuchten Tweets. Es muss also nicht nach der Frage selbst, sondern vielmehr nach den Inhalten möglicher Tweets gesucht werden. Im Rahmen dieser Anwendung soll speziell nach Stichwörtern und Nutzern in deutschsprachigen Tweets gesucht werden.

Die Entscheidung, ob in der Vergangenheit oder in der Gegenwahrt gesucht werden soll bestimmt, welche der APIs zum Einsatz kommen muss. Durch die im vorhergehenden Kapitel gemachte Festlegung, wird die Anwendung nur Tweets aus der Vergangenheit auswerten.

4.1.4 Rohform und Speicherung der Daten

Die Rohform der von der API zurück gelieferten Daten ist textbasiert und entspricht formal dem JSON-Format[19]. Darüber hinaus kann das Rückgabeformat je nach eingesetztem Werkzeug variieren.

Das im Rahmen dieser Arbeit eingesetzte Modul Tweepy ermöglicht den Zugriff auf zurück gelieferte Tweets zum einen in Form eines Strings im JSON-Format, zum anderen in Form

[18] Unter https://developer.twitter.com/en/docs/tweets/search/overview findet sich eine Übersicht über die verschiedenen Filteroptionen.

[19] Unter https://developer.twitter.com/en/docs/tweets/data-dictionary/overview/intro-to-tweet-json.html findet sich eine gute Einführung in die verwendete JSON-Struktur.

einer eigenen API, welche den Zugriff auf die einzelnen Eigenschaften sehr komfortabel gestaltet. Die obige Abbildung 6 zeigt den grundsätzlichen Ablauf.

Da in den folgenden Bearbeitungsschritten, die verfügbaren Daten zumeist sequentiell durchlaufen werden und wenig umfangreich sind, bietet sich für die Speicherung der Tweets das Textformat an, bei dem die einzelnen JSON-Objekte zeilenweise als Strings einer Liste gehalten werden. Dies unterstützt zudem die Intention dieser Arbeit, einen möglichst einfachen Zugang zur Thematik zu öffnen.

4.2 Informationsextraktion

Mit dem Abschluss der Datensammlung im vorhergehenden Schritt liegen nun die für uns relevanten Daten in einer ersten rohen Form vor. Um sie zielführend und ressourcenschonend weiter bearbeiten zu können, bedarf es einer weiteren Aufbereitung der Daten in drei Stufen, welche in der folgenden Abbildung dargestellt ist und in den nächsten Kapiteln weiter ausgeführt werden wird.

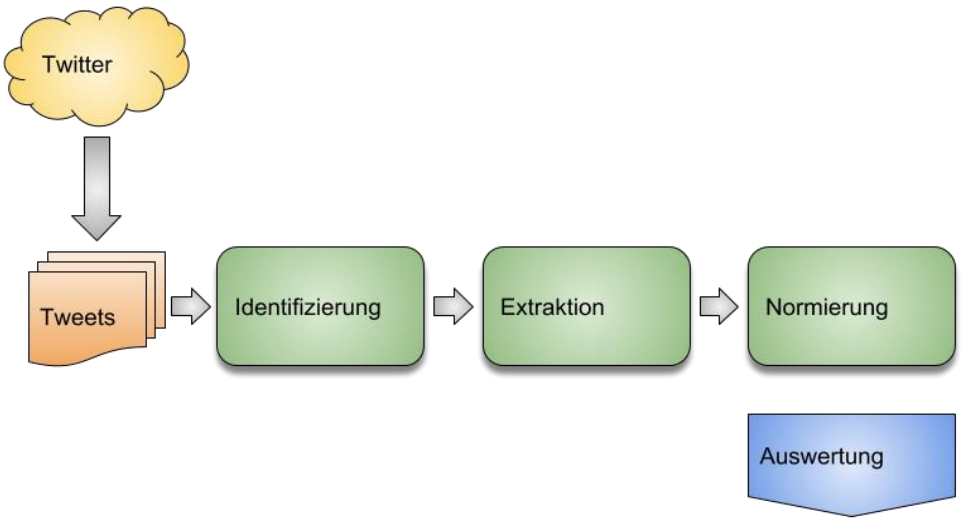


Abbildung 7: Aufbereitung der Daten in drei Stufen, hier durch die grünen Kästen symbolisiert. Nach der Auswertung werden die Daten der Auswertung, hier als blaues Kästchen symbolisiert, zugeführt.

Auch wenn die Basis dieser Arbeit Daten des Dienstes Twitter bilden, so sind die im Bild zu sehenden Stufen Identifizierung, Extraktion sowie Normierung in einer angepassten Form Teil der regelmäßig durchzuführenden Vor- und Aufbereitung textlicher Daten.

4.2.1 Identifizierung der relevanten Daten

Grundlegend für die weitere Verarbeitung ist die Identifizierung der zur Beantwortung der Fragestellung notwendigen Datenbestandteile. Dies impliziert, dass die Anforderungen hieran nicht statisch sind, sondern sich den unterschiedlichen Fragestellungen anpassen müssen.

Im Rahmen dieser Arbeit sollen die folgenden Eigenschaften Berücksichtigung finden:

- die ID des Tweets
- der Zeitstempel des Tweets
- der Nachrichtentext
- die Anzahl der Likes
- die Anzahl der Retweets

Interessant in diesem Zusammenhang wäre die Berücksichtigung der Anzahl an Antworten (Replies) und kommentierten Weiterleitungen (Quote Tweets). Beides ist jedoch nur im Rahmen des kostenpflichtigen Premium bzw. Enterprise Accounts[20] möglich, weshalb hierauf verzichtet wird.

Handelt es sich bei einem Tweet um eine Weiterleitung oder aber eine kommentierte Weiterleitung, so ist der Ursprüngliche Tweet Teil des aktuellen Tweets. Dieser Umstand kann für die Gewinnung zusätzlicher Daten genutzt werden, indem

- der Ursprüngliche Tweet selbst mit aufgenommen wird und
- die Verbindung zwischen den Tweets gehalten wird.

Die Berücksichtigung des ursprünglichen Tweets ermöglicht hierbei Einblicke in den Mechanismus, nach denen sich Tweets verbreiten. Einschränkend muss darauf verwiesen werden, dass der jeweils als Ursprung angegebene Tweet tatsächlich der originäre Tweet ist. Dazwischen liegende Tweets werden nicht berücksichtigt.

4.2.2 Extraktion der relevanten Daten

Bei der hier vorgesehenen Verwendung des Tools Tweepy liegt die Datenbasis als Textdatei vor (vergl. Kapitel 4.1.4 - *Rohform und Speicherung der Daten*). Jede Zeile repräsentiert hierbei ein Tweet im JSON Format. Dies ermöglicht einen direkten Zugriff und Verarbeitung der Daten mit dem standardmäßigen Funktionsumfang von Python. Hierbei wird ein dreistufiger Prozess initialisiert:

- einlesen des Tweets im JSON-Format
- Extraktion der Zieleigenschaft anhand der Vorgaben

[20] Unter https://developer.twitter.com/en/docs/tweets/data-dictionary/overview/tweet-object findet sich eine annotierte Auflistung aller Eigenschaften eines Tweet-Objekts

- sichern der Zieleigenschaften in eine neue Datenbasis

Die unten stehenden Darstellung verdeutlicht nochmals die zuvor ausgeführten Stufen der Extraktion mit dem Ziel, den Umfang der Daten durch Selektion der zur weiteren Auswertung notwendigen Daten deutlich zu verkleinern.

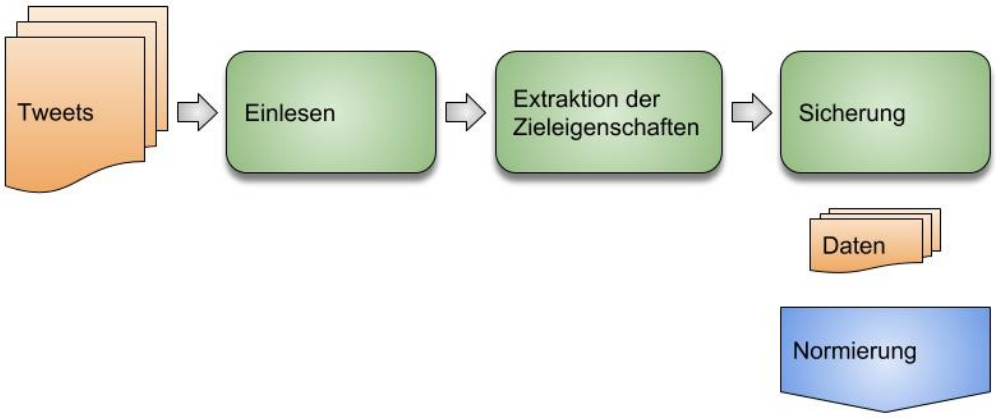


Abbildung 8: Dreistufige Prozesskette für die Extraktion der Daten

Während das Einlesen und Sichern als vorbereitende bzw. abschließende Vorgänge angesehen werden können, bildet der Prozess „Extraktion der Zieleigenschaften“ die Kernfunktionalität ab. Im Rahmen der Ausführung werden hierbei aus der Menge der vorhandenen Eigenschaften genau diejenigen Merkmale des Tweets extrahiert, die zur weiteren Analyse benötigt werden. Die folgende Abbildung verdeutlicht dies nochmals:

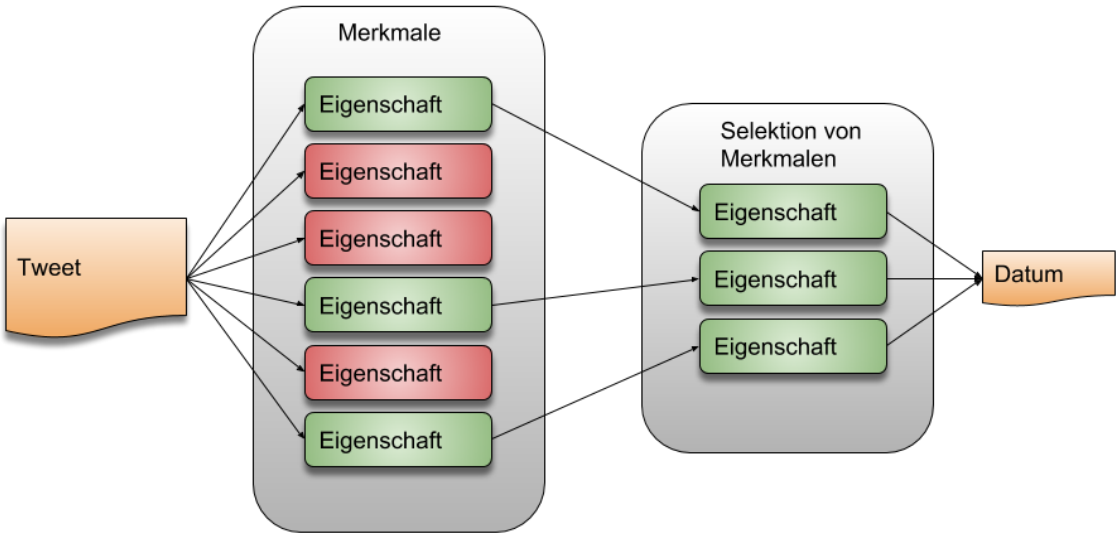


Abbildung 9: Detail der Merkmalsextraktion. Die grünen Eigenschaften symbolisieren die zur Übernahme selektierten Merkmale. Als Ergebnis wird ein deutlich kleinerer und relevanterer Datensatz erhalten.

Als Format des neuen Datensatzes wird wiederum das JSON Format eingesetzt, da es sich, wie bereits erwähnt, um ein Python natives Format handelt und so der Intention dieser Arbeit entgegenkommt.

4.2.3 Normierung der relevanten Daten

Nach der Identifizierung relevanter Eigenschaften und deren Extraktion liegt die für die folgenden Auswertungen relevante Teilmenge an Daten vor. Um die Qualität der Auswertung weiter zu optimieren, werden die gewonnenen Daten weiter normiert.

Aus Gründen der Übersichtlichkeit soll dieser Prozess sequentiell im Anschluss der Datenreduktion durchgeführt werden. Auch hier richtet sich der Umfang und die Qualität an die konkrete Aufgabenstellung und kann wesentlich umfangreicher sein. Im Rahmen dieser Arbeit sollen die folgenden vier Normierungen durchgeführt werden:

- Normierung der Zahlenwerte
- Normierung der Datums- und Zeitangaben
- Normierung der Sprache
- Normierung der Nachrichteninhalte

Das folgende Schaubild verdeutlicht beispielhaft die Umsetzung dieser Aufgabe:

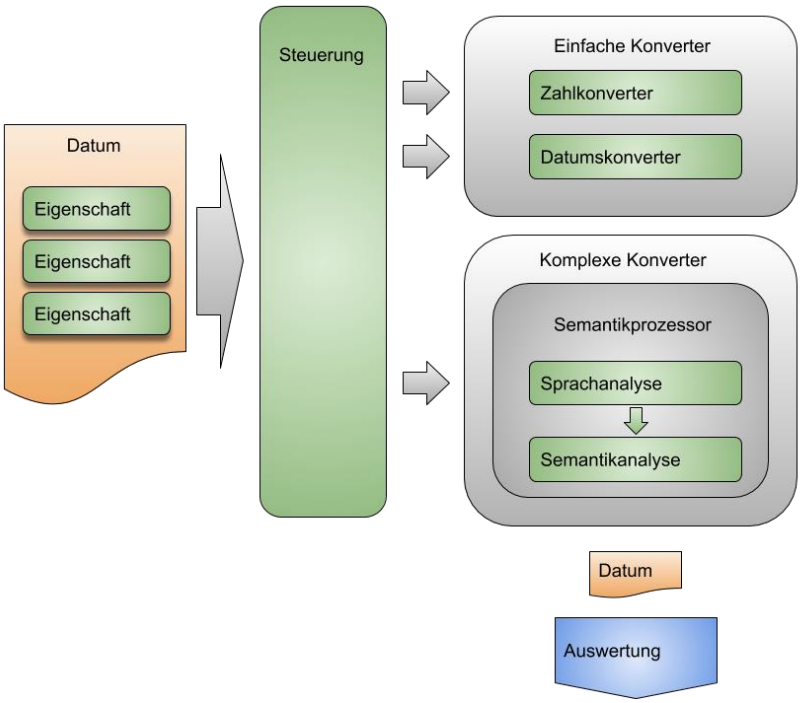


Abbildung 10: Übersicht des Normierungsprozesses. Eine Steuerung steuert die Bearbeitung von Eigenschaften durch Methoden (Konverter). Gruppen mehrerer Methoden können Prozessoren bilden (Komplexe Konverter).

Über eine Steuerung wird jede Eigenschaft einem Konverter zugeführt, der den textlichen Inhalt typisiert. Ein Konverter im Sinne dieser Arbeit ist eine Methode, welche eine definierte Umformung durchführt. Einfache Konverter führen diese Aufgabe direkt und ohne weitere Vor- oder Nachbearbeitung aus. Beispielsweise würde ein Datumskonverter einen Text in ein gültiges Datumsformat umformen.

Komplexe Konverter hingegen benötigen für Ihre Aufgabe verschiedene Vor- und Nachbereitungen, hier durch den Semantikprozessor angedeutet. Ein Prozessor im Rahmen diese Arbeit ist eine zusammengehörende Gruppe von Methoden oder Konvertern, die gemeinsam eine definierte Aufgabe umsetzen.

Aufgabe des Semantikprozessors ist hierbei die Sprach- und Semantikanalyse mit dem Ziel, beides zu normieren. Nachfolgend sollen die beiden letztgenannten Punkte nochmals weiter ausgeführt werden.

4.2.3.1 Normierung der Sprache

Unter der Normierung der Sprache wird in dieser Arbeit die Vereinheitlichung und Reduktion der Sprache verstanden. Um dies umzusetzen, werden die Nachrichtentexte in mehreren Schritten vorverarbeitet. Im Detail sind dies:

- Entfernen nicht deutscher Sätze
- Tokenisierung des Inhaltes
- Entfernen aller nicht gewünschter Bestandteile (Token)
- Entfernen aller deutschen Stopwörter
- Lemmatisierung der verbleibenden Wörter

Besonders problematisch ist hierbei ein nur vereinzelt auftretendes Vorkommen anders sprachlicher Wörter, Abkürzungen und Slang-Ausdrücken. Im Rahmen dieser Arbeit werden zum Zwecke der Einfachheit nicht deutsche Sätze gelöscht und einzelne Vorkommen nicht deutscher Wörter, Abkürzungen etc. ignoriert.

4.2.3.2 Normierung der Nachrichteninhalte

Als Normierung der Nachrichteninhalte wird in dieser Arbeit die Klassifizierung der textlichen Nachricht in eher neutrale, positive oder negative verstanden. Als Basis dient hierfür der zuvor normierte Nachrichtentext.

Die semantische Einteilung selbst erfolgt auf Basis eines bereits vortrainierten Modells. Der Einsatz vortrainierter Modelle für die Umsetzung ist sinnvoll, da dies Vorgehen praxisnah ist und deren vollständige Programmierung und Anlernen den Rahmen und Intention dieser Arbeit übersteigen würde.

4.2.4 Weitere Optimierungen und Anreicherungen

Im Rahmen der zuvor beschriebenen Vorverarbeitung können die vorhandenen Daten weiter angereichert und optimiert werden.

Eine Anreicherung dient der Sicherung von später nicht mehr verfügbaren Parametern oder der Optimierung nachgelagerter Prozesse. Im Rahmen dieser Arbeit gehört hierzu die Länge des ursprünglichen Tweets vor der Optimierung, sowie die Anzahl der informationstragenden Wörter.

Auch die semantische Analyse des Nachrichtentextes und dessen Ersatz durch eine entsprechende Zuordnung zu einer Kategorie kann als eine Anreicherung aufgefasst werden.

Daneben können weitere Optimierungen vorgenommen werden. So kann es durchaus sinnvoll sein, im Rahmen von vorhandenem Domänenwissen einzelne Wörter nicht zu beachten oder aber besonderen Wert beizumessen. Offensichtliche Fehler, Allgemeinplätze oder feststehende Redewendungen können je nach Fragestellung mehr oder weniger relevant sein. Die konkrete Art und Weise ist somit extrem von der konkreten Fragestellung abhängig und bedarf der regelmäßigen Evaluierung. Hierzu gehört auch die Berücksichtigung von Worthäufigkeiten, wie im Kapitel 2.1.2.3 - *Term Frequency – Inverse Document Frequency* beschrieben.

Ziel dieser Arbeit ist die grundlegende Darstellung, weshalb auf weitere Optimierungen verzichtet wird.

4.3 Informationsauswertung

Nach Durchführung der letzten Normierung können die Daten ausgewertet werden. Im Rahmen dieser Arbeit soll dabei auf folgende Eigenschaften eingegangen werden:

- Worthäufigkeit
- Vernetzung der gesammelten Tweets
- Verhältnis Länge zur Anzahl der Retweets
- Verhältnis Länge zur Anzahl der Likes
- Verhältnis Stimmung zu informationstragenden Worten
- Verhältnis Stimmung zu absoluter Länge
- Verhältnis Stimmung zur Anzahl der Retweets
- Verhältnis Stimmung zur Anzahl der Likes

Auch hier liegt der Fokus mehr auf einer beispielhaften Implementierung, als auf einer umfänglichen Auswertung, welche nicht Teil der in dieser Ausarbeitung behandelten Thematik ist.

4.3.1 Worthäufigkeit

Es wird eine Auszählung über die Häufigkeit der vorkommenden Wörter durchgeführt. Durch die Vorverarbeitung werden hierbei nur die Wortstämme berücksichtigt.

Ziel ist der Erhalt eines Überblicks über die zugrunde liegende Basis an Wörtern als Möglichkeit einer ersten Einordnung. Hierzu werden die Wortvorkommen ausgezählt und mit Hilfe eines Streudiagramms und einer Wortwolke visuell dargestellt.

4.3.2 Vernetzung der gesammelten Tweets

Zur Untersuchung der Beziehungen der vorliegenden Tweets untereinander, dient ein Netzwerkgraph. Die Knoten des Graphen entsprechen den Tweets und die Kanten zwischen den Knoten einer Verbindung zwischen zwei Tweets.

Eine Verbindung im Sinne dieser Arbeit besteht zwischen einem Tweet und dem auf ihm beruhenden kommentierten oder nicht kommentierten Retweet wie in der folgenden Abbildung verdeutlicht:

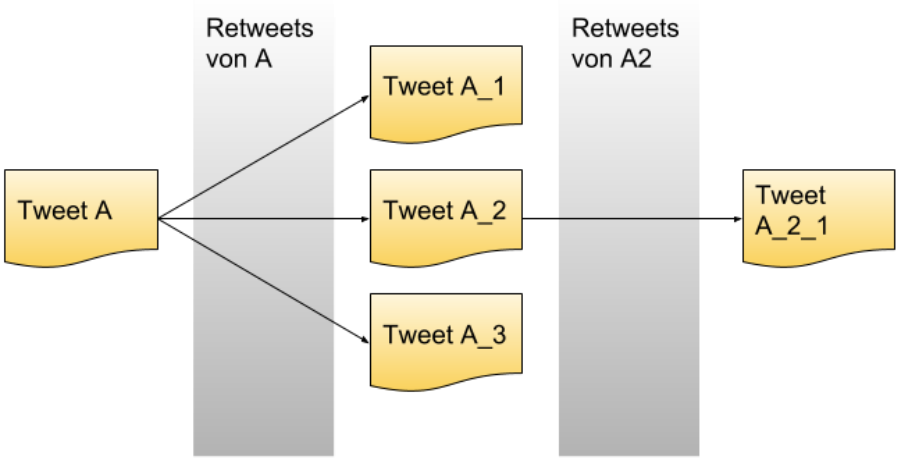


Abbildung 11: Struktur eines Tweets und seiner Retweets

Die Abbildung stellt einen Tweet (Tweet A) dar, der dreimal retweetet wird. Hierdurch entstehen drei neue Tweets (Tweet A_1, A_2 und A_3). Der Tweet A_2 wird ein weiteres Mal geteilt, wodurch wiederum ein neuer Tweet (Tweet A_2_1) entsteht.

Diese Struktur kann als ein Graph aufgefasst werden, dessen Knoten für Tweets und dessen Kanten für eine Verbindung in Form eines Retweets stehen, wie es die folgende Abbildung darstellt.

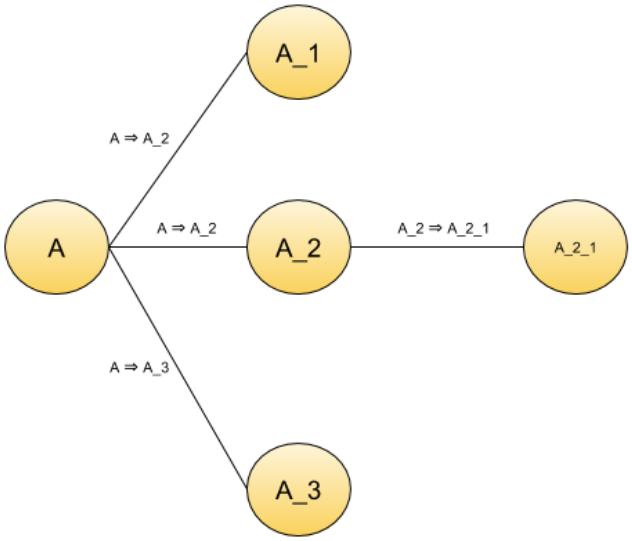


Abbildung 12: Graph eines Tweets und seiner Retweets

Auf dem Bild zu sehen sind die ursprünglichen Tweets, jedoch als Knoten des Graphen. Jede Kante steht für eine Weiterleitung und vercodet in ihrer Beschriftung die Verbindung. So steht A ➔ A_2 für die Aussage: Tweet A_2 ist ein Retweet des Tweets A.

Durch die Darstellung als Graph und der damit einhergehenden Visualisierung lassen sich Rückschlüsse auf den Zusammenhang und die Verbreitung von Nachrichten über Twitter gewinnen. Orte großer Dichte repräsentieren hierbei eine enge Vernetzung oder einen hohen Einfluss einzelner Tweets.

Durch die Einbeziehung der Wertung könnten so Rückschlüsse auf die Verbreitung von Stimmungen gewonnen werden. Dieser Ansatz wir im Rahmen dieser Arbeit jedoch nicht weiterverfolgt.

4.3.3 Betrachtung der Verhältnisse zueinander

Als dritter Punkt sollen einige Verhältnisse zwischen den Eigenschaften der gesammelten Tweets untersucht werden. Da Tendenzen dargestellt werden sollen, kommen Streudiagramme zum Einsatz.

Als erstes wird hierzu die absolute Länge der Tweets in Abhängigkeit zur Häufigkeit gesetzt, mit der ein Tweet geliked oder weitergegeben wurde. Geklärt werden soll hier die Fragestellung, ob die Länge eines Tweets hierauf Einfluss hat.

In einem zweiten Schritt wird die ermittelte Stimmung zum einen in Bezug zur absoluten Länge des Tweets und zum anderen zur Anzahl der sinntragenden Wörter gesetzt. Die Fragestellung hier lautet, ob die Längen einen Einfluss auf die Stimmung der Tweets haben.

Als letztes wird die von Twitter ermittelte Anzahl der Likes und Retweets in Bezug zur Stimmung gesetzt. Auch hier ist die Fragestellung, ob die Stimmung die anderen Faktoren beeinflusst hat.

4.4 Darstellung und Halten der Ergebnismenge

Die im Rahmen dieser Anwendung entstehende Arbeit dient nicht dem produktiven Einsatz, sondern der Vermittlung von Kenntnissen. Aus diesem Grunde ist eine dauerhafte Haltung und Wiederverwendung der entstandenen Zwischen- und Ergebnismenge keine primäre Anforderung.

Um dennoch Ergebnisse auch längerfristig vorhalten zu können, wird eine Möglichkeit geschaffen, die Daten als Textdateien zu speichern und diese lokal downloaden zu können. Dies gilt auch für die im Rahmen der Auswertung ausgegebenen Grafiken, welche in geeigneter Form exportierbar sein sollen.

5 Konkrete Umsetzung

Nachdem im vorhergehenden Kapitel die umzusetzende Anwendung von der konzeptionellen Seite aus diskutiert wurde, soll im Folgenden die konkrete Umsetzung und die hier getroffenen finalen Entscheidungen betrachtet werden.

Zu besseren Kenntlichmachung sind die Namen von Kapiteln, Objekten, Methoden und Dateien der Anwendung in einer **`abweichenden Formatierung, wie hier zu sehen,`** dargestellt. Eine ausführliche Aufstellung der verwendeten Module kann der Anwendung entnommen werden.

5.1 Anwendung

Als Basis der hier beschriebenen Anwendung dient Colaboratory (Colab) von Google. Hierbei handelt es sich um ein im Umfeld von TensorFlow angesiedeltes Jupyter Notebook.

Colab zeichnet sich durch seinen einfachen Zugang, der Möglichkeiten zur Veröffentlichung sowie der bereits vorinstallierten Tensor Flow Unterstützung besonders aus. Bei Bedarf können weitere Module sowie Bibliotheken hinzugefügt werden.

Im Rahmen dieser Arbeit wird der Begriff Anwendung als Synonym, für das mit Colab erstellte Notebook verwendet.

5.1.1 Session

Eine Session oder auch Sitzung beschreibt im Rahmen dieser Anwendung eine zeitliche Periode, in der mit der Anwendung gearbeitet wird und diese mit der Laufzeitumgebung verbunden ist.

Innerhalb einer Session bleiben die Ergebnisse einmal durchgeführter Aktionen bestehen und sind innerhalb der Anwendung aufgrund des interaktiven Charakters verfügbar. Insbesondere gilt dies für Installationen, Importe und der Definition von Methoden.

Wird innerhalb einer bestimmten Zeitspanne keine Aktivität festgestellt, so unterbricht die Anwendung in einem ersten Schritt die Verbindung zur Laufzeitumgebung. Nach Ablauf einer weiteren Frist werden die Ressourcen der Anwendung recycelt. Sämtliche Anpassungen und Ergebnisse, die nicht das Notebook selbst betreffen, gehen hierdurch verloren. Insbesondere betrifft dies durchgeführte Uploads, erstellte Dateien und Installationen.

Dieser Vorgang kann innerhalb eines Notebooks auch manuell über das Menü angestoßen werden.

5.1.2 Fehlerbehandlung

Die Aufgabe der Fehlerbehandlung ist die Behebung von behebbaren Fehlern und die Überführung einer Anwendung in einen sicheren Zustand bei nicht behebbaren Fehlern. Zudem soll dem Nutzer eine sinnvolle Information über die Art des Problems gegeben werden.

Ziel der im Rahmen dieser Arbeit vorliegenden Anwendung ist die Vermittlung von Wissen. Nutzer sollen ermutigt werden, sich eine Kopie des Notebooks zu erstellen und damit aktiv zu arbeiten. Da es sich um ein interaktives System handelt, können die einzelnen Schritte direkt und mehrmals angestoßen, sowie der Code direkt geändert werden.

Eine herkömmliche Fehlerabfangung wäre daher nicht zielführend, besonders, da Colab selbst über eine sehr ausführliche Fehlerverfolgung und -anzeige verfügt, die sehr gezielt eingesetzt werden kann. Es wird daher grundsätzlich auf Fehlerabfangungen im Code verzichtet.

5.1.3 Kommentierung

Die vorliegende Anwendung führt den Nutzer exemplarisch durch die umgesetzte Lösung. Neben ausführbaren Code enthält die Anwendung hierzu umfangreiche Textpassagen zur Beschreibung des Vorgehens.

Aus diesem Grunde wird daher bis auf wenige Ausnahmen grundsätzlich auf eine Dokumentierung innerhalb des Codes verzichtet.

5.1.4 Strukturen innerhalb von Colab

Die vorliegende Anwendung nutzt die natürlichen Strukturelemente eines Notebooks, auf welche kurz eingegangen werden soll[21]. Ein Notebook besteht hierbei aus den folgenden drei Komponenten, die frei verwendet werden können:

- Textblock – Ein in sich geschlossener Bereich mit Text. Eingegebene Texte können mit einem einfachen Markup[22] formatiert werden, um Abschnitte zu strukturieren. Hierdurch ist eine Gliederung des Notebooks als Ganzes möglich.
- Codeblock – Ein in sich geschlossener Bereich, der ausführbaren Code enthält und manuell ausgeführt werden muss. Innerhalb eines Codeblocks kann auf die Inhalte anderer Blöcke, zuvor installierter Module oder gesetzter Variablen und Konstanten zugegriffen werden. Methoden innerhalb von Codeblöcken müssen vor der ersten Verwendung einmal ausgeführt worden sein.

[21] Unter https://colab.research.google.com/notebooks/basic_features_overview.ipynb findet sich ein guter Einstieg hierzu.

[22] Unter https://colab.research.google.com/notebooks/markdown_guide.ipynb findet sich ein guter Einstieg hierzu.

- Formular – Eine Möglichkeit, einfache Eingabeelemente mittels einer Markupsprache[23] zu einem Code hinzuzufügen. Sie bieten die Möglichkeit, Code zu verbergen und erhöhen damit die Übersichtlichkeit. Zwischen beiden Ansichten kann schnell mittels eines Doppelklicks der linken Maustaste gewechselt werden. Im Rahmen dieser Anwendung wird der Begriff Formular synonym für den dahinterliegenden Code verwendet.

Somit unterscheidet sich der Aufbau eines Notebooks grundsätzlich von dem einer herkömmlichen Anwendung und erinnert eher an ein Skript. Insbesondere ist eine eventgesteuerte Programmierung nicht möglich.

5.1.5 Strukturierung der Anwendung

Auf der obersten Ebene unterteilt sich die vorliegende Anwendung in sieben Schritte, die strukturiert durch die Aufgabenstellung der Anwendung führen. Jeder Schritt enthält wiederum erklärenden und begleitenden Text sowie Formulare mit ausführbarem Code.

Im Folgenden wird daher nur grundsätzlich auf die Struktur der Anwendung eingegangen und explizit auf die dort enthaltenen Ausführungen verwiesen.

5.1.5.1 Schritt 1: Initialisierung der Anwendung

Im Rahmen der Initialisierung werden notwendige Installationen, Downloads und Importe durchgeführt. Dieser Schritt teilt sich hierbei auf zwei Formulare auf:

- **`Notwendige Installationen ausführen`** – Ausführung aller notwenigen Installationen, teilweise unter Vorgabe der Version.
- **`Notwendige Importe, Autorisierungen und Downloads ausführen`** – Ausführung aller notwendigen Importe, Download der benötigten Datenkorpusse sowie die Erstellung des Wörterbuchs **`auth_help`** mit allen für Twitter benötigten Token. Diese können über die Eingabefelder festgelegt werden.

Aufgrund des nur temporären Bestehens des Notebooks, müssen alle hier durchgeführten Aktionen zu Beginn jeder neuen Session erneut ausgeführt werden.

5.1.5.2 Schritt 2: Datenhaltung einrichten

In diesem Schritt werden alle für die Anwendung relevanten Aktionen zum Halten der Daten durchgeführt. Den Kern bilden hierbei fünf Listen, die mit dem Formular **`Aktuelle Daten ausgeben`**, initialisiert werden:

- **`raw_tweets`** – Liste unbearbeiteter Tweets
- **`reduced_tweets`** – Liste der reduzierten und unbearbeitete Tweets
- **`preprocessed_tweets`** – Liste der vorverarbeiteten Tweets

[23] Unter https://colab.research.google.com/notebooks/forms.ipynb findet sich ein guter Einstieg hierzu.

- **`tweets_relations`** – Liste der extrahierten Relationen
- **`word_list`** – Liste der in den Tweets vorkommenden Wörter

Innerhalb der Listen wird ein Tweet als JSON-formatierter String gehalten, Relationen und Wörter als Wörterbücher. Im folgendem wird hierauf Bezug genommen, wenn auf Speicher bzw. dem Speichern von Daten eingegangen wird.

Im weiteren Verlauf werden mehrere Hilfsfunktionen definiert, um den Inhalt der Listen einfach ausgeben und löschen zu können. Nach der erstmaligen Ausführung können diese in der gesamten Anwendung genutzt werden.

5.1.5.3 Schritt 3: Modellerstellung für die semantische Analyse

Ein zentraler Punkt der hier vorliegenden Arbeit ist die semantische Analyse und die damit einhergehende Klassifizierung der jeweils vorliegenden Tweets in eher negative, neutrale sowie positive, wie in 4.2.3.2 - *Normierung der Nachrichteninhalte* angesprochen.

Für die Analyse wird ein vortrainiertes Modell auf Basis eines **`DNNClassifier`** Objekts sowie ein Korpus deutscher Google News (**`nnlm-de-dim128/1`**[24]) verwendet. Ein **`DNNClassifier`** ist ein für die Klassifizierung anhand von gegebenen Labels geeignetes Objekt innerhalb von TensorFlow. Um das so erhaltene Modell besser auf Twitter abstimmen zu können, erfolgt ein weiteres Training auf Basis klassifizierter, deutscher Tweets.

Aufgrund der Regeln für die Nutzung von Twitter, darf ein allgemein verfügbarer Korpus mit Tweets nur die IDs beinhalten, nicht jedoch die Nachrichten selbst. Im Rahmen dieser Arbeit wurde daher ein Korpus klassifizierter Tweets[25] verwendet und anhand der IDs im Zeitraum vom 21. bis zum 26.12.2018 um die Nachrichtentexte ergänzt. Für das eigentliche Training wurden hiervon je 800 positiv und negativ bewertete Tweets, sowie 1600 neutral bewertete Tweets verwendet und in der Datei **`trainset_final.csv`** verfügbar gemacht.

Für die Arbeit mit dem im Rahmen dieser Anwendung erstellten Notebook bedeutet dies, dass diese Datei im Vorfeld von dem Benutzer selbst erstellt werden muss. Die hierfür notwendigen Informationen und Quellen sind zusammen mit einem Mustercode Teil der Anwendung[26].

Aufgrund der langen Zeitspanne kann nicht ausgeschlossen werden, dass Teile der Tweets nicht mehr der ursprünglichen Klassifizierung entsprechen.

Um die Übersichtlichkeit zu erhöhen, teilt sich der aktuelle Schritt in vier Abschnitt auf:

---

[24] Weitere Informationen finden sich unter https://tfhub.dev/google/nnlm-de-dim128/1.

[25] Insgesamt umfasst der Korpus 6704 noch existierende Datensätzen. Hiervon wurden 995 als positiv, 4262 als neutral sowie 1447 als negativ bewertet. Weitere Informationen und Download unter https://www.spinningbytes.com/resources/germansentiment/.

[26] Zusätzlich findet sich ein Mustercode im Anhang dieser Ausarbeitung im Kapitel *Code zur Anreicherung von Tweets mit Text anhand deren IDs*.

- **Zusätzliche Trainingsdaten aus Twitter vorbereiten** – Formulare für die Vorbereitung der klassifizierten Daten von Twitter für das Training. Hierfür muss die Datei **trainset_final.csv** zuvor hochgeladen werden.
- **Modell erstellen** – Formulare für das Erstellen des eigentlichen Modells auf Basis eines vortrainierten Modells
- **Training mit zusätzlichen Daten aus Twitter und Genauigkeit testen** – Formulare zur Durchführung eines zusätzlichen Trainings anhand der klassifizierten Daten aus Twitter. Hierfür muss die Datei **trainset_final.csv** zuvor hochgeladen werden.
- **Nutzen des Modells** – Formulare, die ein Test und die Nutzung des erzeugten Modells erlauben.

5.1.5.4 Schritt 4: Datenbeschaffung

Im vierten Schritt wird auf die Ziehung der Daten von Twitter eingegangen (siehe hierzu auch 4.1 - *Twitter als Datenquelle*). Im Mittelpunkt steht hierbei das Modul Tweepy, welches die Abfragen objektbasiert kapselt.

Zentraler Bestandteil ist das Formular **Twitter API abfragen**. Über dessen Eingabefeldern können alle Parameter der Suche festgelegt werden:

- **Suchbegriff** – ein Suchbegriff oder eine Liste mit Suchbegriffen
- **Nutzer** – ein Nutzername
- **Anzahl** – Die maximale Anzahl der zurückgelieferten Tweets. Die Angabe einer Anzahl begrenzt hierbei die Rückgabe auf ein Maximum, garantiert aber nicht eine Mindestanzahl.
- **Typ** – Als mögliche Typen kann zwischen User (nutzt die Angabe unter Nutzer) und Suchbegriff (nutzt die Angabe unter Suchbegriff) gewählt werden.

Innerhalb des Formulars erfolgt die Umsetzung durch zwei Funktionen. Die Funktion **grap_by_user** nutz die User Timeline API[27] von Twitter und gibt Tweets des übergebenen Nutzers wider. Die Funktion **grab_by_query** nutz hingegen die Search API[28] und gibt Tweets anhand kommaseparierter Suchbegriffe zurück. Beiden Methoden benötigen die zentral hinterlegten Anmeldeinformationen und schreiben ihre Ergebnisse direkt in **raw_tweets**, wo sie zur weiteren Verarbeitung vorliegen.

[27] Unter https://developer.twitter.com/en/docs/tweets/timelines/api-reference/get-statuses-user_timeline.html findet sich eine Auflistung aller möglichen Parameter.

[28] Unter https://developer.twitter.com/en/docs/tweets/search/api-reference/get-search-tweets.html findet sich eine Auflistung aller möglichen Parameter.

Über den Parameter **`tweet_mode = „extended“`** wird die Rückgabe von Tweets mit Texten über 140 Zeichen unterstützt.

5.1.5.5 Schritt 5: Datenvorverarbeitung

Die Datenvorverarbeitung ist eine weitere zentrale Aufgabe und beinhaltet die Selektion, Vorbereitung und Normalisierung der relevanten Daten aus den vorliegenden Rohdaten. Im Rahmen dieser Anwendung geschieht dies in zwei Schritten:

5.1.5.5.1 Datenselektion und Datenreduktion

Im ersten Schritt wird durch die in **`raw_tweets`** vorhandenen Einträge iteriert und hierbei jeweils die relevanten Einzeldaten selektiert. Dies entspricht von der Konzeption her dem in den Kapiteln 4.2.1 und 4.2.2 diskutierten Vorgehen. Die so selektierten Daten werden im JSON Format in **`reduced_tweets`** abgelegt. Der entsprechende Code befindet sich im Formular **`Datenselektion und -reduktion durchführen`**. Im Rahmen dieser Arbeit werden die folgenden Eigenschaften berücksichtigt:

- das Datum der Erzeugung
- die ID des Tweets
- der Text des Tweets
- die Anzahl der Retweets
- die Anzahl der Likes
- der Typ des Tweets

Twitter[29] unterscheidet bei der Weitergabe von Tweets zwischen einem einfachen Retweet sowie einem Quote Tweet. Ein Quote Tweet wird im Gegensatz zum Retweet vor dem weitergegeben kommentiert.

Ein Retweet erkennt man an der Existenz eines Knoten **`retweeted_status`**, ein Quote Tweet an der Existenz eines Knoten **`quoted_status`**. Ein Tweet kann hierbei über beide Knoten verfügen. Die Knoten beinhalten Informationen über den ursprünglichen Tweet.

Die zuvor beschriebene Struktur nutzt die Anwendung, um in Anlehnung an Kapitel 4.3.2 die Liste **`tweets_relations`** zu befüllen und so eine Auflistung von zusammenhängenden Tweets zu generieren. Zudem werden die ursprünglichen Tweets ebenfalls in die Ergebnismenge mit aufgenommen.

[29] Unter https://developer.twitter.com/en/docs/tweets/data-dictionary/overview/intro-to-tweet-json findet sich eine vertiefende Einführung zur JSON Struktur eines Tweets.

5.1.5.5.2 Normalisierung der Daten und semantische Analyse

In einem zweiten Schritt findet die in Abschnitt 4.2.3 diskutierte Normierung der nun in **reduced_tweets** vorliegenden Daten statt. Dies geschieht im Formular **Normalisierung und semantische Analyse durchführen**.

Innerhalb des Formulars dient hierbei die Methode **run_normalization** als Steuerung, welche die nachgelagerten Normierungen in Form einzelner Methodenaufrufe steuert. Diese Methoden entsprechend den im oben genannten Kapitel angeführten, einfachen und komplexen Konvertern, welche an dieser Stelle nicht weiter getrennt werden. Folgende Normalisierungen kommen zur Anwendung:

- Sprachfilter (Methode **language_filter**) – Selektion und Entfernung von Tweets nicht deutscher Sprache
- Tokenisierung (Methode **token_filter**) – Tokenisierung der enthaltenen Texte und Entfernen von Hash- und At-Tags sowie Links
- semantische Analyse der Tweets (Methode **run_sentimentanalyse**) – Anreicherung der Tweets um ein Stimmungsmerkmal
- Datumskonvertierung (Methode **date_converter**) – Entfernen von Texten aus dem Datumsstring
- Entfernung von Stopwörter (Methode **stopword_filter**) – Entfernen aller Stopwörter
- Lemmatisierung (Methode **lemmatizer**) – Umwandlung der vorhandenen Wörter in deren Grundform

Der Zeitpunkt der semantischen Analyse ist hierbei ein Kompromiss an das verwendete vortrainierte Modul. Dessen Korpus wurde an nicht vollständig normalisierten Daten anhand von Google News trainiert. Als Vorverarbeitung fand lediglich eine leerzeichenbasierte Tokenisierung statt.

Die abschließende Lemmatisierung erfolgt mit Hilfe des Moduls SpaCy, da das Natural Language Toolkit (NL-Toolkit) hier keine geeignete Lösung für die deutsche Sprache anbietet.

Zum Abschluss der Normalisierung werden die nun in ihrer Grundform (Lemma) vorliegenden Wörter gezählt und das Ergebnis wortweise in **word_list** geschrieben.

Als Ergebnis der Normierung liegen die ursprünglich gezogenen Daten in einer wesentlich kompakteren und nunmehr vergleichbaren Form vor und werden zur weiteren Verwendung in **preprocessed_tweets** gesichert.

5.1.5.6 Schritt 6: Datenauswertung

Im sechsten Schritt rückt die Auswertung der in den vorhergehenden Schritten vorbereiteten Daten, wie in Kapitel 4.3 diskutiert, in den Fokus.

Zum Einsatz kommen hauptsächlich interaktive Streudiagramme von Altair. Diese haben den Vorteil, dass sie verschoben und in Ihrer Skalierung geändert werden können. Zudem verfügen sie über eigene Menüs zur Speicherung oder aber Weiterverarbeitung mittels des Vega Editors[30]. Hierbei handelt es sich um eine Grammatik für interaktive Grafiken.

Streu- oder aber auch Scatterdiagramme haben die Besonderheit, eher einen Trend, denn einer genauen Wiedergabe von Einzelwerten zu dienen. Einzelne Werte können in der Regel nicht aus ihnen extrahiert werden, wohingegen Tendenzen sehr deutlich erscheinen.

Die Zahlenbasis, der folgenden Diagramme basiert auf eine Abfrage mit dem Suchwort „Tsunami“ vom 27.12.2018. Als Ergebnis wurden 2000 Tweets zurückgegeben. Nach der Vorverarbeitung ergab dies 3246 Tweets mit 1328 Relationen. Insgesamt existierten 2429 unterschiedliche Wörter.

Um die Übersichtlichkeit zu erhöhen, wurde auch dieser Schritt in mehrere Abschnitte aufgeteilt:

- Erstellen der Datenbasis
- Worthäufigkeit
- Relationen der Tweets untereinander
- weitere grafische Auswertungen

5.1.5.6.1 Erstellen der Datenbasis

Zu Darstellung der Grafiken verwendet die Anwendung als zentrale Objekte zwei DataFrame Objekte des Python Moduls Pandas, welche alle relevanten Daten beinhalten:

- **`data_frame`** – alle für die Streudiagramme notwendigen Werte
- **`word_data_frame`** – Daten für die Verwendung innerhalb einer Wortwolke.

Neben dem Formular **`Datenbasis für die Auswertung erstellen`**, bieten die verbleibenden zwei Formulare Ausgabemöglichkeiten zur Kontrolle der Daten an.

5.1.5.6.2 Worthäufigkeit

Wie bereits in Kapitel 4.3.1 - *Worthäufigkeit* erwähnt, kann diese als ein erstes Stimmungsbild des Ergebnisses angesehen werden. Innerhalb dieser Anwendung wird die absolute Häufigkeit des Auftretens eines Wortes als Maß verwendet und auf zwei Arten dargestellt.

Zum einen erfolgt die Darstellung mit Hilfe eines Streudiagramms. Innerhalb des Formulars **`Übersicht über die Worthäufigkeit (absolut) anzeigen`** kann hierbei über den

[30] Tiefer gehende Informationen zu Vega Lite finden sich unter https://vega.github.io/vega-lite.

Eingabeparameter **`Grenzwert`** gesteuert werden, ab welchem Vorkommen ein Wort berücksichtigt wird.

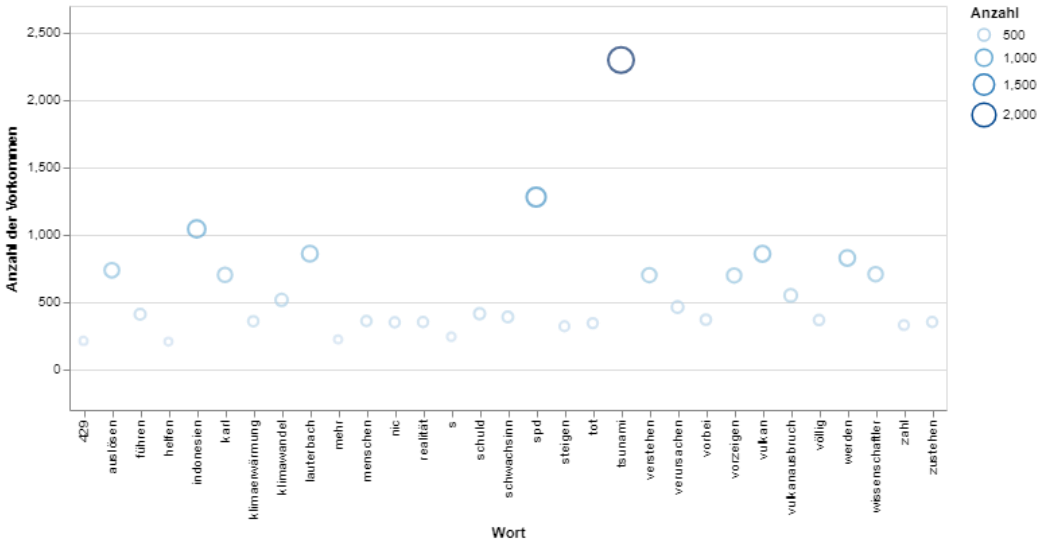


Abbildung 13: Absolute Häufigkeit eines Wortes ab einer Anzahl von 250. Die Größe und Farbe der Kreise geben hierbei einen Rückschluss auf die Anzahl der Nennungen

Zum anderen wird für die Visualisierung mit dem Formular **`Word Cloud der vorkommenden Wörter anzeigen`** eine Wortwolke verwendet. Hierbei werden die berücksichtigten Wörter in einem Bild angeordnet. Je nach Konfiguration ergeben Größe oder Farbe Rückschlüsse auf deren Vorkommen. Über das verbleibende Formular kann die erzeugte Datei heruntergeladen werden.

Abbildung 14: Visualisierung aller vorkommenden Wörter innerhalb einer Word Cloud. Größe und Färbung lassen einen Rückschluss auf deren Vorkommen zu.

Neben der Verwendung aller vorkommenden Wörter, kann es sinnvoll sein, die ursprünglichen Suchwörter auszuschließen und nicht die absolute Häufigkeit, sondern einen statistischen

Wert, basierend auf deren Vorkommen und Typ als Basis zu verwenden (vergl. hierzu auch 2.1.2.3). Dieser Ansatz wird hier nicht weiterverfolgt.

5.1.5.6.3 Relationen der Tweets untereinander

Für die Darstellung der Verbindungen von Tweets untereinander wird ein Graph verwendet und zur Ausgabe gebracht (vergl. hierzu auch 4.3.2 - *Vernetzung der gesammelten Tweets*). Für die Erstellung ist das Formular **`Relationengraph anzeigen`** verantwortlich.

Die Verwendung eines Graphen kann Rückschlüsse auf die Art der Verbreitung von Tweets der Ergebnismenge geben. Diese kann eher homogen sein, so dass sich zumeist 1:1 Beziehungen ergeben, oder aber eher zu einer Haufenbildung mit wenigen 1:n Beziehungen neigen. Hierbei handelt es sich um Tweets, die sehr oft geteilt wurden und die so einen erhöhten Einfluss haben, wobei die Gründe vielfältig sein können.

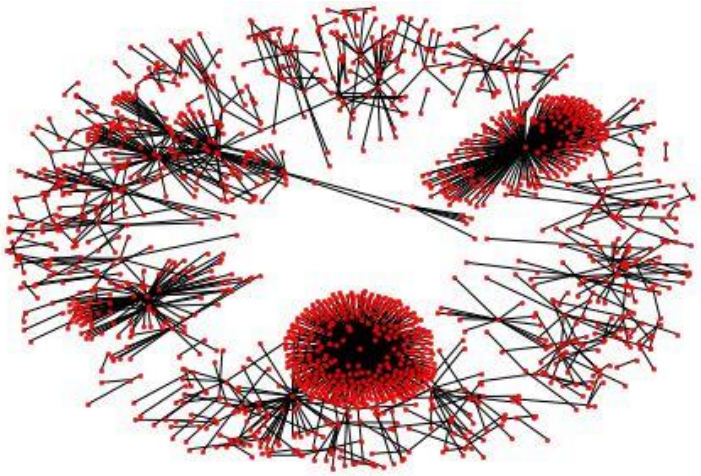


Abbildung 15: Relationen von Tweets. Deutlich sind zwei große sowie zwei kleinere Haufenbildungen zu erkennen. Diese zeigen eine starke Verbreitung einzelner Tweets.

In der obigen Abbildung fallen besonders zwei Gebiete mit einer starken Haufenbildung unten und rechts oben auf. Ursache sind wenige Tweets, welche sehr häufig geteilt wurden. Bei dem Rest handelt es sich zumeist um eine normale Verbreitung.

Mit Hilfe der verbleibenden zwei Formulare lässt sich der Graph als Bild sowie als Datei für das Programm Gephi[31] ausgeben.

5.1.5.6.4 Weitere grafische Auswertungen

Im letzten Teil der grafischen Auswertung, werden über die drei Formulare

- **`Übersicht absolute Länge zur Anzahl der Likes und Retweets anzeigen`**

[31] Bei dem Programm Gephi handelt es sich um ein Open Source Programm für die Visualisierung von Graphen. Es kann unter https://gephi.org/ heruntergeladen werden.

- **Übersicht Stimmung zur Länge der Tweets anzeigen**
- **Übersicht Stimmung zur Anzahl der Likes und Retweets anzeigen**

jeweils entsprechende Streudiagramme, entsprechend der Konzeption unter 4.3.3 - *Betrachtung der Verhältnisse zueinander*, ausgegeben. Auf diese soll im Folgenden kurz eingegangen werden.

5.1.5.6.4.1 Absolute Länge zur Anzahl der Likes und Retweets

Diese Übersicht zeigt den Zusammenhang zwischen der absoluten Länge inklusive aller Füllwörter, Links etc. zur Anzahl der Likes und Retweets. Zusätzlich geben die Größe und Farbe der Kreise Rückschlüsse auf die Anzahl der Tweets sowie deren Stimmung als weitere Dimensionen wie im folgenden Diagramm zu sehen ist.

Grün steht hierbei für eine eher positive, rot für eine eher negative Stimmung. Die Kreisgröße ist proportional zur Anzahl der im Diagramm adressierten Tweets.

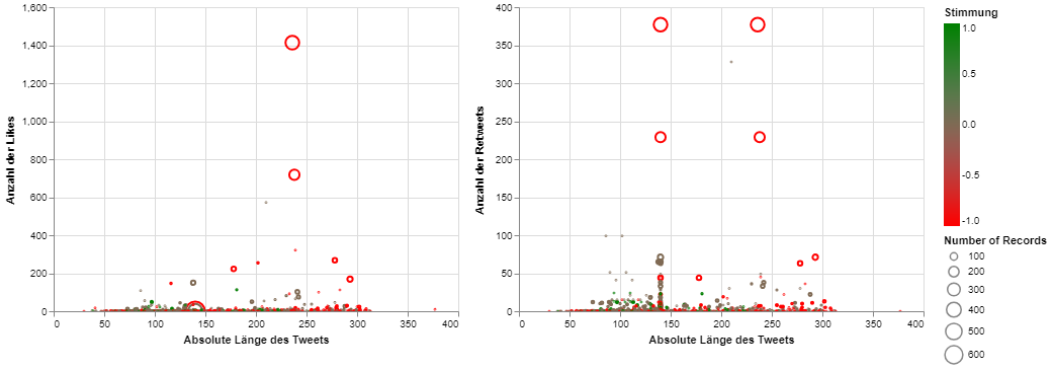


Abbildung 16: Absolute Länge von Tweets zur Anzahl der Likes und Retweets. Innerhalb des Diagramms bilden Größe und Färbung der Kreise weitere Dimensionen ab.

Die Fragestellung hier lautet somit, ob zwischen den genannten Eigenschaften absolute Länge zur Anzahl der Likes bzw. Retweets ein Zusammenhang existiert.

5.1.5.6.4.2 Übersicht Stimmung zur Länge des Tweets anzeigen

Bei dieser Übersicht wird die Stimmung des Tweets zur absoluten Länge und zur Anzahl der informationstragenden Wörter in Verbindung gesetzt.

Hierbei steht der Wert -1 für eine eher negative, 0 für eine eher neutrale und 1 für eine eher positive Stimmung. Deutlich ist in den folgenden Diagrammen die Dominanz negativer Tweets über der gesamten Längenskala zu erkennen.

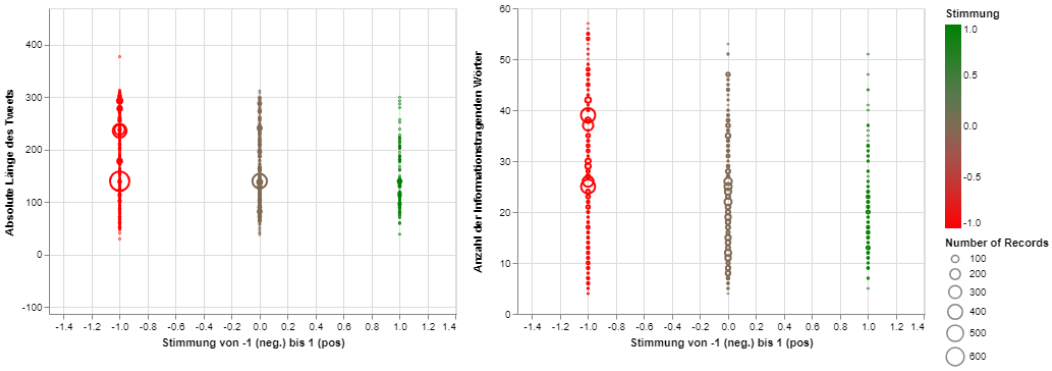


Abbildung 17: Stimmung von Tweets zu deren Länge. Auch hier bilden die Größe und Farbe der Kreise weitere Dimensionen ab.

Die Farbe und Größe der Kreise ist wie zuvor beschrieben kodiert, was bezogen auf die Farbcodierung redundant ist. Trotz dessen wird hierdurch die Analyse intuitiver.

Die Fragestellung hier lautet, ob grundsätzlich längere Tweets zu einer bestimmten Stimmung hintendieren, oder aber, ob ein Zusammenhang mit der Anzahl informationstragender Wörter gefunden werden kann.

5.1.5.6.4.3 Übersicht Stimmung zur Anzahl der Likes und Retweets

Abschließend wird die Stimmung des Tweets zur Anzahl der Likes und Retweets dargestellt. Als weitere Dimensionen tauchen hier die Länge der Tweets sowie die Anzahl der Vorkommen, wie im folgenden Diagramm zu sehen ist, auf.

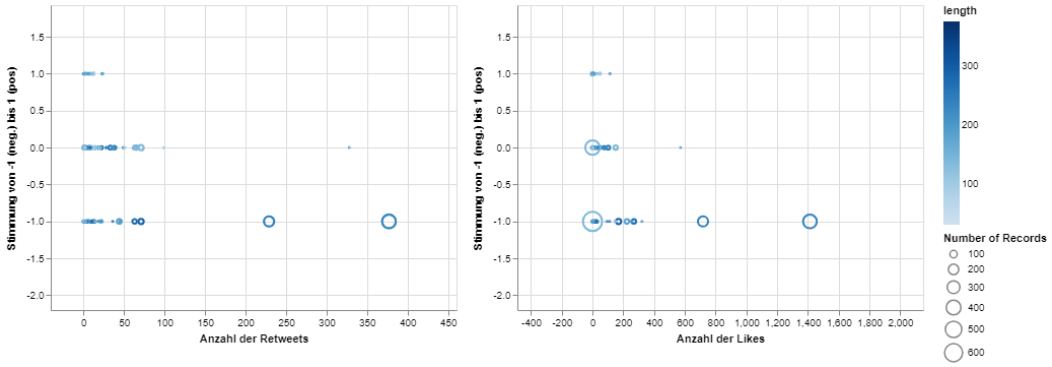


Abbildung 18: Stimmung von Tweets zu der Anzahl an Likes und Retweets. Analog zu den oberen Diagrammen bilden die Größe und Helligkeit hier weitere Dimensionen ab.

Die Intensität der blauen Farbe der Kreise ist hierbei proportional zu deren Länge. Die Größe der Kreise zu der Anzahl der vertretenden Tweets.

Die Fragestellung hier lautet, ob eher positive, oder aber eher negative Tweets weitergeleitet werden.

5.1.5.7 Schritt 7: Datenspeicherung, Download und Wiederherstellung der Rohdaten

Als letzter Schritt der Anwendung werden einige Möglichkeiten vorgestellt, wie mit Daten innerhalb von Colab umgegangen werden kann. Hierzu gehört der Export der vorliegenden Daten, wie in Abschnitt 4.4 gefordert.

Hierzu wurden Formulare mit jeweils eigenem Schwerpunkt angelegt, der exemplarisch behandelt wird:

- **`Download aller Daten`** – Zeigt beispielhaft den Download aller gespeicherten Daten als eigenständige Dateien. Hierzu werden die entsprechenden Dateien erzeugt, innerhalb von Colab gespeichert und dann heruntergeladen.
- **`Beispiel zum Laden und verwenden der erstellten Dateien`** – Zeigt beispielhaft anhand der vorbereiteten Tweets sowie der Wortliste den Zugriff auf gespeicherte Daten. Hierzu müssen diese Dateien zuvor erzeugt worden sein.
- **`Mit Google Drive verbinden`** – Zeigt das notwendige Vorgehen, um auf eigene, in Google Drive gespeicherte Daten, von Colab aus zuzugreifen zu können.
- **`Codeschnipsel zur Anreicherung von Tweet-IDs`** – Zeigt einen Mustercode, der verwendet werden kann, um anhand einer Tweet ID den zugehörigen Text des Tweets zu erhalten.

5.1.5.8 Weiterführende Links

Als letzten Abschnitt der Anwendung wurde eine Liste mit einer Auswahl interessanter Links als Ergänzung und Vertiefung der behandelten Thematik hinzugefügt. Diese erhebt keinen Anspruch auf Vollständigkeit oder Repräsentativität.

5.2 Konfiguration

In diesem Abschnitt soll kurz auf die notwendigen Voraussetzungen für den Betrieb der Anwendung sowie dem Vorgehen bei deren Inbetriebnahme eingegangen werden.

Eine umfassende Einweisung kann jedoch im Rahmen dieser Arbeit nicht geleistet werden. Hier sei auf die entsprechenden Dokumentationen verwiesen.

5.2.1 Voraussetzungen

Für die im Rahmen dieser Arbeit erstellten Anwendung wird deren Einsatz innerhalb von Google Drive empfohlen. In diesem Fall besteht auf Grund der browserbasierten Ausführung keine Notwenigkeit zur lokalen Installation weiterer Software, jedoch ist eine aktive Internetverbindung notwendig.

5.2.1.1 Twitteraccount

Für die Nutzung der Daten von Twitter wird ein persönlicher Account[32] bei Twitter benötigt. Innerhalb dieses Accounts muss weiterhin eine Anwendung registriert[33] sein. Während die persönlichen Angaben zum eigentlichen Account valide sein müssen, reichen grundlegende Angaben für die Anmeldung einer Anwendung aus.

Im Anhang befinden sich Screenshots der realen Anmeldung dieser Anwendung zur Information. Grundsätzlich können die hier gemachten Angaben und Schlüssel jederzeit geändert werden.

5.2.1.2 Google Drive

Für die Arbeit mit Colaboratory wird ein Account[34] bei Google Drive vorausgesetzt. Bei Google Drive handelt es sich um einen vom Suchmaschinenbetreiber Google angebotenen Onlinespeicher, der kostenlos bis zu 15 GB freien Speicher sowie eine eigene Infrastruktur für eine Vielzahl eigener Anwendungen sowie Drittanwendungen bereitstellt.

5.2.1.3 Colaboratory

Colaboratory (Colab) ist ein auf Jupyter Notebook basierendes Notebook, dass auf Google Drive gehostet wird. Um Colab innerhalb von Google Drive nutzen zu können, muss die Anwendung als neuer Anwendungstyp verknüpft[35] werden, sofern dies noch nicht geschehen ist.

Sobald die Anwendung mit dem Account verknüpft wurde, können neue Notebooks angelegt und vorhandene lokale und externe Notebooks geöffnet werden.

5.2.1.4 Lokales Ausführen der Anwendung

Soll entgegen des hier präferierten Vorgehens lokal gearbeitet werden, so muss eine Laufzeitumgebung für Python Notebooks[36] sowie ein Python 3 Interpreter vorhanden sein. In diesem Fall sind die noch fehlenden Bibliotheken manuell zu installieren.

Besonders bei dem Einsatz der Grafikbibliothek Altair kann die Notwendigkeit entstehen, diese anzupassen oder in Gänze zu ersetzen.

5.2.1.5 Erstellen der Trainingsdatei trainset_final.csv

Zur Erstellung der Datei **`trainset_final.csv`** (vergl. 5.1.5.3) ist keine spezielle Software, sondern nur die entsprechende Datei mit den IDs und deren Klassifizierung notwendig.

---

[32] Unter https://twitter.com/i/flow/signup kann die Registrierung für Twitter erfolgen.
[33] Unter https://developer.twitter.com/en/apply-for-access kann eine Anwendung registriert werden.
[34] Unter https://www.google.com/drive/ kann ein Account bei Google Drive angelegt werden.
[35] Innerhalb von Google Drive befindet sich im Menü für die Neuerstellung von Dokumenten ein entsprechender Eintrag.
[36] Unter https://www.anaconda.com/ findet sich mit Anaconda eine geeignete Umgebung.

Wird der im Anhang und innerhalb der Anwendung verfügbare Mustercode zur Zuordnung von Texten zu den IDs von Tweets verwendet, wird ein Python 3 Interpreter, das Modul Tweepy, ein aktiver Twitteraccount sowie eine aktive Internetverbindung benötigt.

5.2.2 Öffnen und Ausführen der Anwendung

Die eigentliche Ausführung und Distribution der Anwendung können auf verschiedenen Wegen erfolgen, welche im Folgenden kurz erörtert werden. Prinzipiell kann die vorliegende Anwendung diese nutzen.

5.2.2.1 Öffnen und Ausführen der Anwendung als geteilter Link

Als einfachste Variante kann die Nutzung einer Freigabe in Google Drive gesehen werden. Hierfür wird innerhalb von Google Drive die Anwendung für einen bestimmten Nutzer oder aber für alle mit einem gültigen Link freigegeben. Weitere Arbeiten sind in diesem Fall nicht notwendig und der gegebene Link führt direkt zum Notebook.

Dieses Vorgehen hat den Nachteil, dass dem Nutzer hierdurch weitgehende Rechte eingeräumt werden. Gerade mit Blick auf die Forderung, aktiv an dem Dokument zu arbeiten, um zu lernen, sollte dieser Ansatz in der Regel nicht verwendet werden.

5.2.2.2 Öffnen und Ausführen der Anwendung als Upload

Sofern die im vorhergehenden Abschnitt genannten Vorgaben erfüllt sind, kann innerhalb von Google Drive einfach ein neues Colab Notebook erzeugt und die Quelldatei der hier vorliegenden Anwendung hochgeladen werden.

Hierfür existiert im Menü von Colab je ein Menüpunkt für das Öffnen bzw. den Upload eines bestehenden Notebooks. Mögliche Speicherorte sind Google Drive, GitHub sowie der lokale Rechner.

Die Verteilung der Anwendung kann hierbei sowohl in Form einer einfachen Textdatei erfolgen, welche den Quellcode der Anwendung enthält, als auch direkt über GitHub. Ein Account bei GitHub ist hierfür nicht notwendig.

6 Empirische Untersuchung der Umsetzung

Zum Abschluss dieser Arbeit soll die vorliegende Anwendung mit Blick auf die Datenziehung anhand von Suchbegriffen sowie der semantischen Analyse näher untersucht werden.

Aufgrund der vorliegenden Umsetzung als Notebook scheiden übliche Methoden zur Qualitäts- und Fehlerkontrolle, wie automatisierte Testverfahren und die Auswertung von Metriken aus oder sind nur sehr aufwendig umzusetzen. Zudem zielen die Umsetzung und Strukturierung des Codes mehr auf ein einfaches Verständnis ab, denn auf eine robuste Umsetzung für den produktiven Einsatz. Diese Zielsetzung steht jedoch dem Einsatz der zuvor genannten Methoden entgegen.

Im Folgenden wird daher anhand einer begrenzten Datenmenge empirisch untersucht, ob diese den Suchkriterien entsprechen und semantisch korrekt eingeordnet wurden. Hierfür wurden am 27.12.2018 mit dem Suchwort „Weihnachten“ 10 Tweets von Twitter gezogen.

Nach der Vorverarbeitung führte dies zu 15 Tweets mit 5 Relationen und 107 Wörtern, welche im Folgenden untersucht werden. Die Originaltexte finden sich auszugsweise als Teil des Anhangs im Kapitel *Wiedergabe der originalen Texte und Urtexte der Testdaten* wieder.

Auf die Überprüfung der korrekten visuellen Darstellung der Daten wird im Rahmen dieser Arbeit verzichtet.

6.1 Vorhandensein des Suchwortes

In 11 von 15 Tweets ist das gegebene Suchwort in den Texten der Tweets vorhanden. Dies entspricht mehr als 70 Prozent der gezogenen Menge. In den verbleibenden vier Tweets taucht das gesuchte Wort im Hashtag bzw. dem ursprünglichen Text auf.

Somit passen alle gezogenen Tweets zum vorgegebenen Suchwort. Die Prüfung fand durch eine manuelle Kontrolle anhand der Originaldaten statt.

6.2 Semantische Analyse

Im Fokus der folgenden Prüfung steht die semantische Analyse. Hierfür wird zunächst die Genauigkeit des Modells selbst geprüft, sowie im Anschluss die Klassifizierung der zuvor gezogenen Testdaten empirisch geprüft.

6.2.1 Prüfung der Genauigkeit

Zur Prüfung von Modellen zur Klassifizierung bietet TensorFlow hierfür geeignete Methoden an, mit deren Hilfe anhand von bereits klassifizierten Daten die Genauigkeit eines Modells bestimmt werden kann.

Mit Hilfe der Methode **`confusion_matrix`**[37] von Tensor Flow sowie dem Modul Seaborn zur Visualisierung wurde für das in dieser Arbeit verwendete Modell vor und nach dem zusätzlichen Training mit dem Korpus von Twitter (siehe 5.1.5.3 *Schritt 3: Modellerstellung für die semantische Analyse*) je eine Heatmap erstellt. Als klassifizierte Daten kamen bei der Prüfung die zuvor verwendeten Trainingsdaten zum Einsatz.

Bei der vor dem Training erstellten Heatmap ist zu sehen, dass sich die gegebene Klassifizierung deutlich von der vorhergesagten unterscheidet und zu einer eher positiven Beurteilung neigt. Erst nach dem Training sieht man im zweiten Bild eine deutliche Erhöhung der Trefferquote, wie es anhand der nach dem Training ausgegebenen Genauigkeit von über 77 Prozent zu erwarten ist.

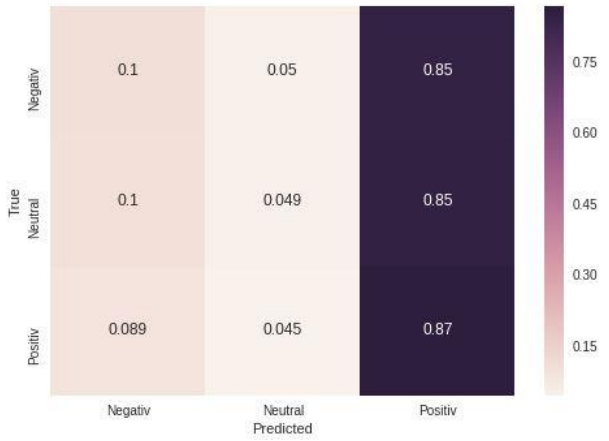


Abbildung 19: HeatMap vor dem ergänzenden Training. Zu sehen ist eine Häufung fehlerhaft als positiv klassifizierten Trainingsdaten.

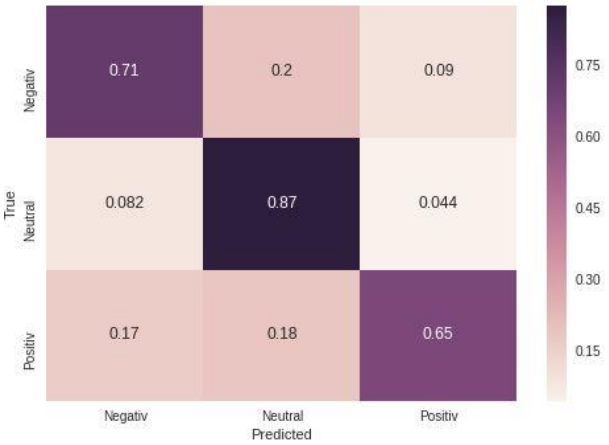


Abbildung 20: HeatMap nach dem ergänzenden Training. Zu sehen ist die gehäuft auftretende Übereinstimmung von vorgegebener und vorhergesagter Klassifizierung.

[37] Unter der Adresse https://www.tensorflow.org/api_docs/python/tf/confusion_matrix findet sich eine vertiefende Beschreibung der Methode.

6.2.2 Empirische Prüfung der Ergebnismenge

Im Folgenden wird die durchgeführte automatisierte Klassifizierung der Tweets nochmal einer empirischen Prüfung unterzogen. Hierbei soll manuell entschieden werden, ob deren Einstufung richtig oder falsch ist.

Als neutral wurden die folgenden Tweets eingestuft:

1. „vater fliegt weihnachten bei stewardess mit der flugbegleiterinnenbegleiter“
2. „der überwiegend islamische irak hat weihnachten auf wunsch der christlichen minderheit zum gesetzlichen feiertag erklärt die“
3. „bei wissen sie auch nach noch ganz genau wie man seine zielgruppe direkt anspricht“

Als positiv wurden die folgenden Tweets eingestuft:

1. „elftes gebot schicke weihnachten niemandem gifs den du das ganze jahr ignoriert hast“
2. „guten morgen allerseits wir hoffen ihr habt weihnachten gut überstanden“
3. „meine eltern haben mir zu weihnachten einen alpaka wanderung geschenkt und es war das beste geschenk aller zeiten“
4. „wurde in somalia verboten wir sind ein muslimisches land und es gibt null toleranz für solche unislam“
5. „und das soll euch als zeichen dienen ihr werdet ein kind das in windeln gewickelt in einer krippe liegt lukas 2 12“

Als negativ wurden die folgenden Tweets eingestuft:

1. „der überwiegend islamische irak hat weihnachten auf wunsch der christlichen minderheit zum gesetzlichen feiertag erklärt die weltreligion islam ist offenbar pluraler und vielschichtiger als die rechtsextremen hassprediger es uns so oft glauben machen wollen“
2. „alle jahre wieder leider auch dieses weihnachten mussten wir teils erheblich betrunkene autofahrer aus dem verkehr ziehen so u a am heiligen abend auf der bei neustadt glewe als wir bei einem 38 jährigen einen atemalkoholwert von 2 23 promille feststellten“
3. „erstes weihnachten ohne jens wer ist jens trauer herrscht für mich 400 tote wieder durch einen zunami was soll diese unnütze berichterstattung soviel leid auf dieser welt“
4. „wurde in somalia verboten wir sind ein muslimisches land und es gibt null toleranz für solche unislamischen feiern in unserem land das kann uns hier nicht passieren wir haben ja einen toleranten islam“

Bei der Begutachtung der Tweets fällt Im Besonderen der offensichtlich positiv gemeinte erste Satz der als negativ eingestuften Tweets auf. Eine Erklärung hierfür könnte der Gebrauch von „rechtsextremen hassprediger“ in einem nicht erkannten positiven Kontext sein.

Ähnlich verhält es sich auch bei dem vierten Satz in diesem Block. Hier fällt auf, dass die gekürzte Variante (Satz vier im Block der positiven Tweets) als eher positiv eingestuft wird. Eine Erklärung könnte die falsch interpretierte Verbindung von „verboten“ und „null toleranz“ im ersteren sein.

Weitergehend fällt auf, dass die gemachten Einteilungen überwiegend in der gegebenen Art Bestand haben können, auch wenn durchaus kontroverse Meinungen bei der Begutachtung durch reale Personen möglich sind. Zudem ist zu bemerken, dass den vorliegenden Texten zumeist keine starke Stimmung innewohnt. Dies erschwert die Beurteilung.

6.3 Fazit

Als Fazit bleibt zu sagen, dass die Anwendung in der Tendenz wie erwartet arbeitet. In allen gezogenen Tweets war das als Suchbegriff genannte Wort vertreten, bei über 70 Prozent der Tweets direkt im Text. Die Ziehung über den Nutzernamen sowie die Visualisierung war nicht Teil der Prüfung.

Die durchgeführte Kategorisierung arbeitet ebenfalls in der Tendenz wie erwartet. Fehlerhafte Kategorisierungen sind möglich und müssen bei der Bewertung der Ergebnisse Berücksichtigung finden. Durch Änderungen am Modell und dem Training könnten hier Verbesserungen erzielt werden.

Ergänzend muss erwähnt werden, dass die untersuchten Texte durchaus Spielraum für deren Einteilung bieten, was eine Klassifizierung erschwert.

7 Zusammenfassung und Ausblick

Ziel dieser Arbeit war die Erstellung eines Python Notebooks, welches einfach verfügbar ist, exemplarisch das Vorgehen bei der Datengewinnung aus sozialen Netzwerken, dem „Social Data Mining“, vorstellt sowie die Möglichkeit bietet, aktiv in den Code einzugreifen.

Hierzu wurden nach einer kurzen Einführung in die Grundlagen zunächst die Anforderungen an einer solchen Anwendung definiert. Im weiteren Verlauf wurden dann die notwendigen konzeptionellen Überlegungen und abschließend die finalen Entscheidungen im Rahmen der Umsetzung erläutert. Fußnoten zeigen mögliche Quellen, um sich tiefergehender als es hier möglich ist, mit der Materie zu beschäftigen und geben weitere Informationen.

Die Arbeit schließt mit einer empirischen Begutachtung der Datenbeschaffung und der darauf basierenden semantischen Analyse ab. Die Akkuranz der Vorhersage mit rund 77 Prozent sowie die sich ergebenen Unstimmigkeiten bei der manuellen Kontrolle der Testdaten zeigt deutlich, dass hier Potential zur Verbesserung existiert. Als mögliche Schritte sind hier vor allem drei zu nennen:

- Nochmalige Evaluierung der Trainingsdaten sowie eine Vergrößerung des Korpus. Der aktuell verwendete Korpus stammt aus 2017 und die bei Twitter verwendete Sprachgewohnheiten könnten sich geändert haben.
- Wegfall des auf Google News basierenden, vortrainierten Modells oder Ersatz zu Gunsten eines geeigneteren. Hierbei würden sich die Anforderungen an das Training entsprechend vergrößern.
- Evaluierung des Zeitpunktes der semantischen Analyse. Die hier zu klärende Fragestellung ist, ob ein vollnormierter Text in einer höheren Genauigkeit bei der Vorhersage resultiert, als ein im Original belassener Text.

Trotz der zuvor genannten Defizite bietet die vorliegende Anwendung einen einfachen, ersten Einblick in einen möglichen Workflow von der Datenbeschaffung über die Vorverarbeitung bis hin zur Auswertung. Es werden Möglichkeiten zur Umsetzung aufgezeigt und der Einsatz etablierter Bibliotheken und Frameworks wie Tensor Flow, Pandas oder dem Natural Language Toolkit demonstriert.

Um dieses Ziel zu erreichen, wurde bewusst bei der Programmierung nicht auf eine möglichst gute und wiederverwendbare Aufteilung hingearbeitet, sondern auf ein möglichst einfaches Verständnis, welches durch Erörterungen innerhalb des Notebooks ergänzt wird.

Literatur- und Quellenverzeichnis

Autoren ApfelWiki (29.03.2009) Python – apfelwiki.de. http://www.apfelwiki.de/Main/Python. Zugegriffen: 25. September 2018

Hope T, Resheff YS, Lieder I (2018) Einführung in TensorFlow; Deep-Learning-Systeme programmieren, trainieren, skalieren und deployen. O'Reilly, Heidelberg

IPython (23.07.2018) The IPython Notebook — IPython 3.2.1 documentation. http://ipython.org/ipython-doc/dev/notebook/notebook.html. Zugegriffen: 27. September 2018

Jannaschk K (2017) Infrastruktur für ein Data Mining Design Framework

Müller AC, Guido S (2017) Einführung in Machine Learning mit Python; Praxiswissen Data Science. O'Reilly, Heidelberg

Pfaffenberger F (2016) Twitter als Basis wissenschaftlicher Studien: Eine Bewertung gängiger Erhebungs- und Analysemethoden der Twitter-Forschung. Springer, s.l.

Piazza F (Hrsg) (2010) Data Mining im Personalmanagement; Eine Analyse des Einsatzpotenzials zur Entscheidungsunterstützung. Gabler, Wiesbaden

Runkler TA (2015) Data Mining; Modelle und Algorithmen intelligenter Datenanalyse. Springer Vieweg, Wiesbaden

Steyer R (2018) Programmierung in Python. Springer Fachmedien Wiesbaden, Wiesbaden

Twitter inc. (2018) Investor Fact Sheet; Q2'18. https://investor.twitterinc.com/static-files/b2037053-e45c-421a-a91e-5271a433f90f. Zugegriffen: 11. November 2018

Wikipedia-Autoren sV (20.09.2018) OAuth. https://de.wikipedia.org/w/index.php?title=OAuth&oldid=180770581. Zugegriffen: 22. September 2018

Wikipedia-Autoren sV (24.09.2018) CPython. https://de.wikipedia.org/w/index.php?oldid=180855847. Zugegriffen: 25. September 2018

Wikipedia-Autoren sV (03.10.2018) Tf-idf-Maß. https://de.wikipedia.org/w/index.php?oldid=175531817. Zugegriffen: 18. Oktober 2018

Abbildungsverzeichnis

Datenverzeichnis

Als Basis für Erstellung der Datei trainset_final.csv mit klassifizierten Tweets diente:

A Twitter Corpus and Benchmark Resources for German Sentiment Analysis. by Mark Cieliebak, Jan Deriu, Fatih Uzdilli, and Dominic Egger. In “Proceedings of the 4th International Workshop on Natural Language Processing for Social Media (SocialNLP 2017)”, Valencia, Spain, 2017

Verfügbar unter https://www.spinningbytes.com/resources/germansentiment/, Zugegriffen: 11. Januar 2019

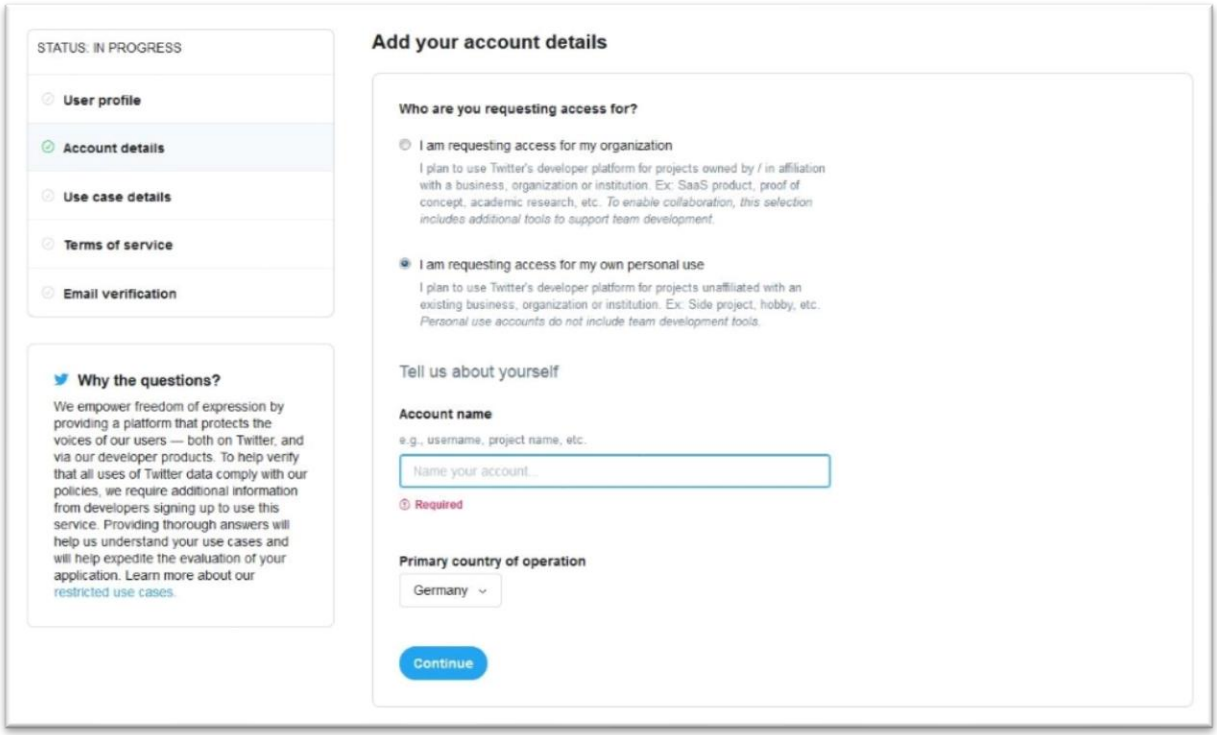
Formelverzeichnis

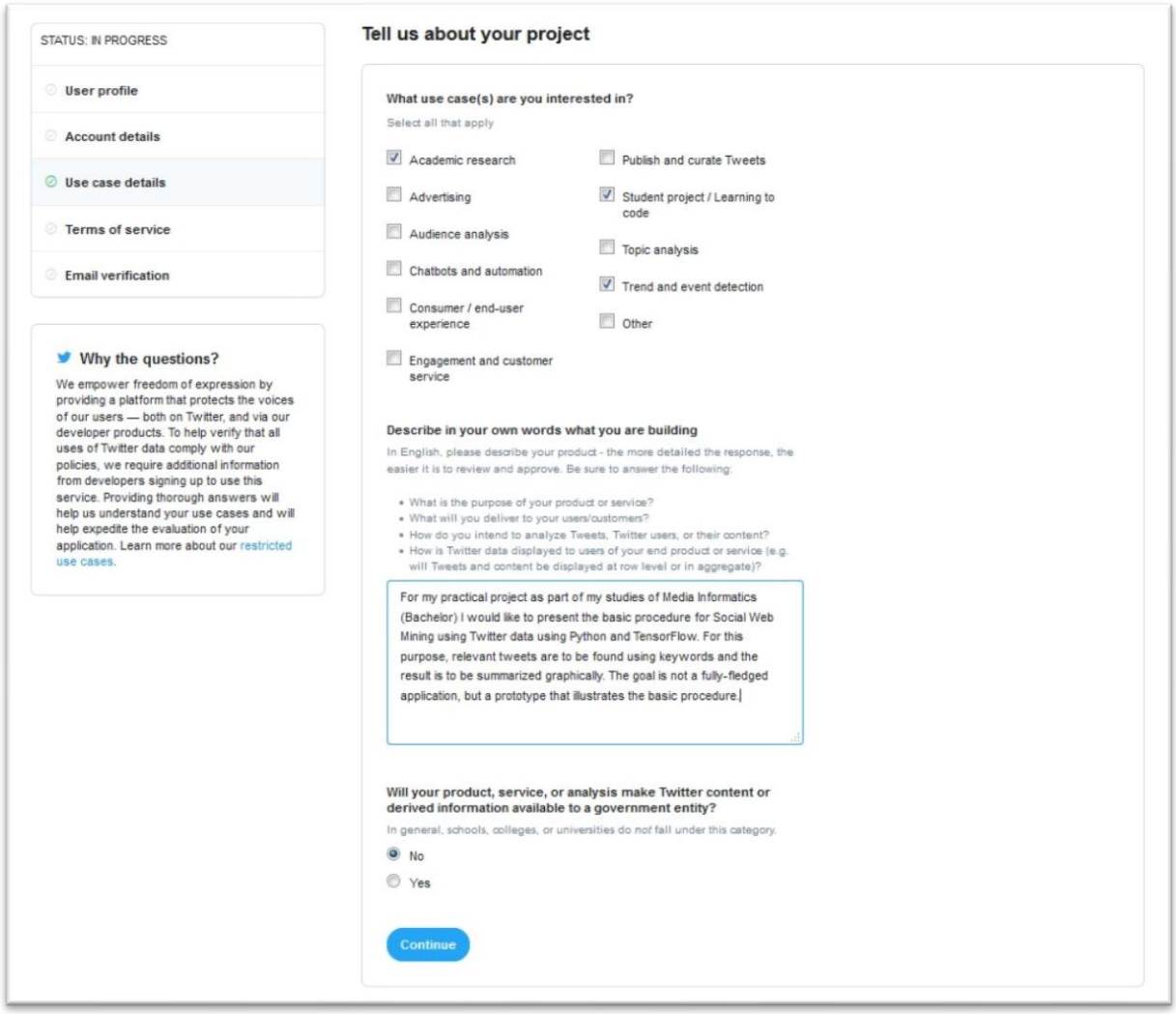
Anhang

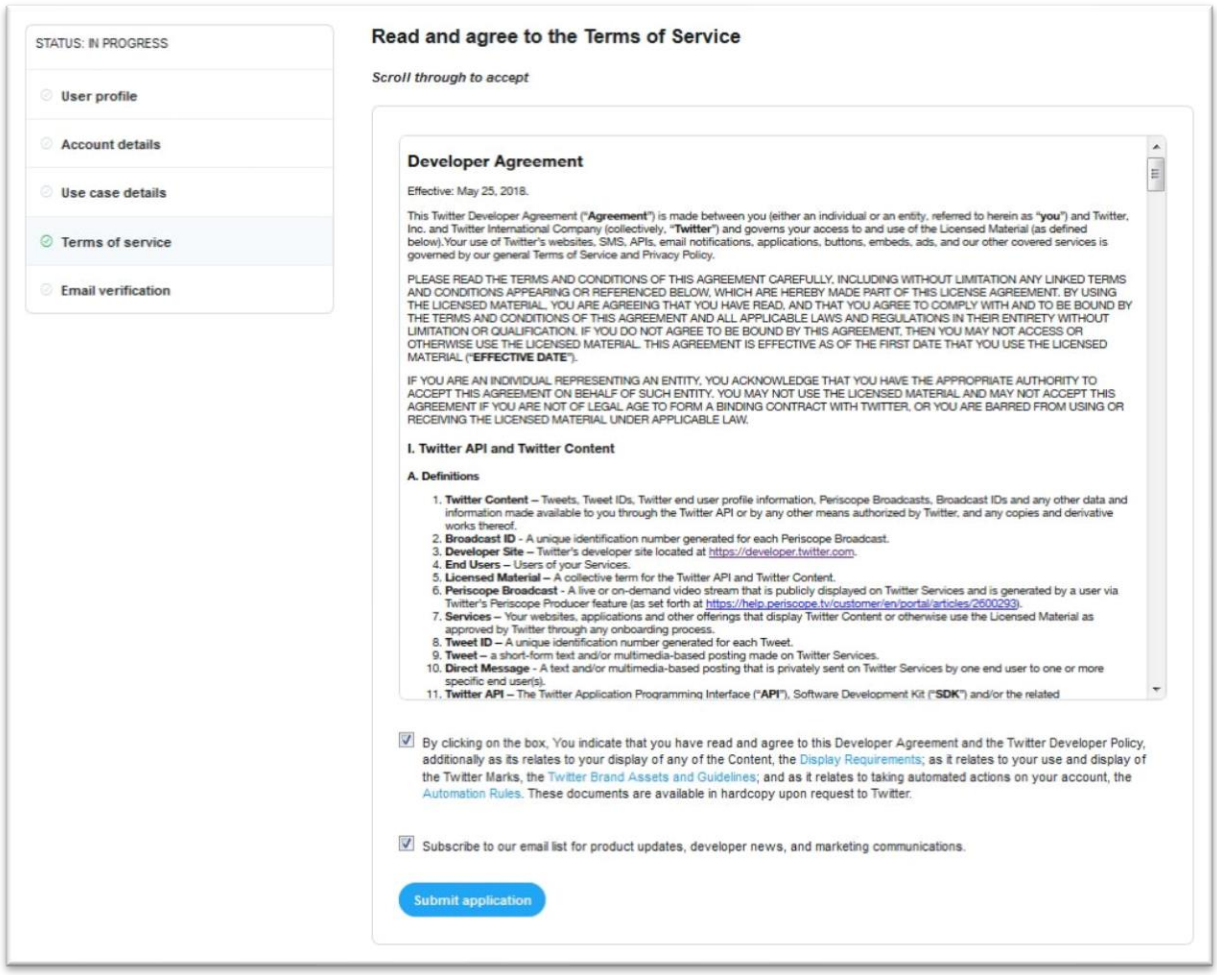
(1) Anmeldevorgang in Twitter unter https://developer.twitter.com/en/apply-for-access

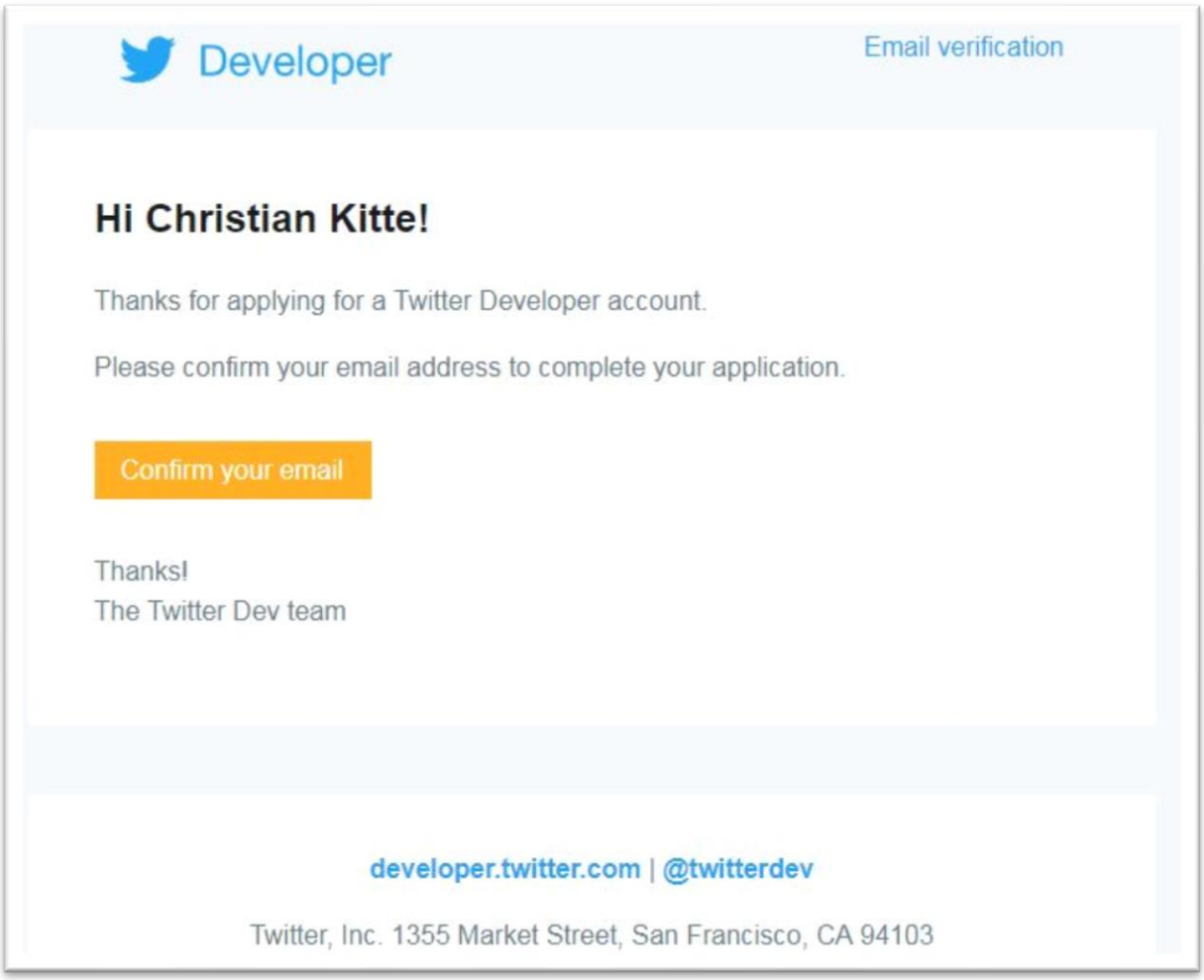
Im Folgenden wird die Anmeldung einer Anwendung zu einem bestehenden Account als Folge von Screenshots wiedergegeben, um einen Eindruck hiervon zu vermitteln. Die Nutzerführung ist hierbei eindeutig, so dass auf einer weiter gehenden Kommentierung an dieser Stelle verzichtet werden kann.

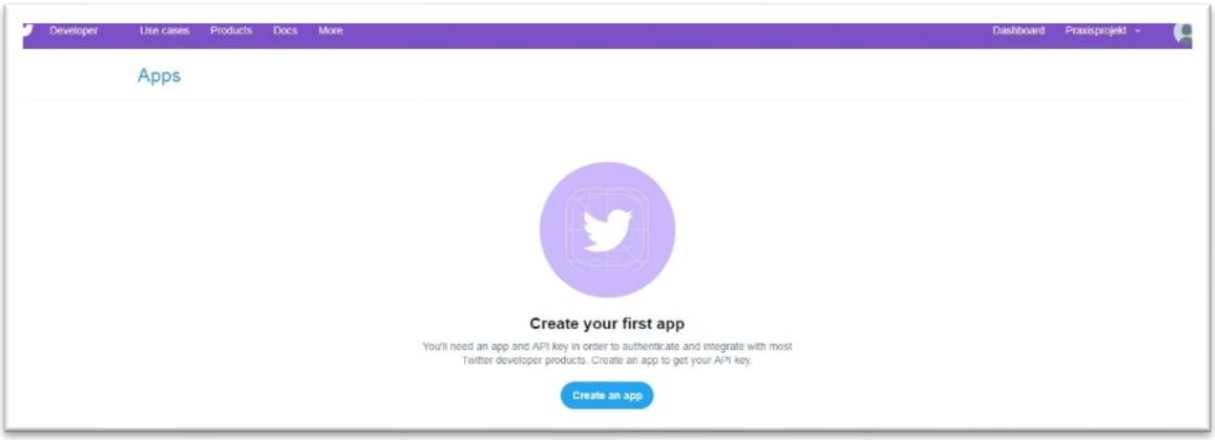
Im Verlaufe des Anmeldeprozesses sind eine Reihe von Eingaben über die geplante Anwendung zu machen. Da die Registrierung einer Anwendung recht zügig geschehen und zudem jederzeit abgeändert werden kann, sollte man nicht zu viel Zeit zum Finden der geeignetsten Angaben investieren. Gerade bei dem ersten Kontakt mit Twitter kann dies sinnvoll sein.

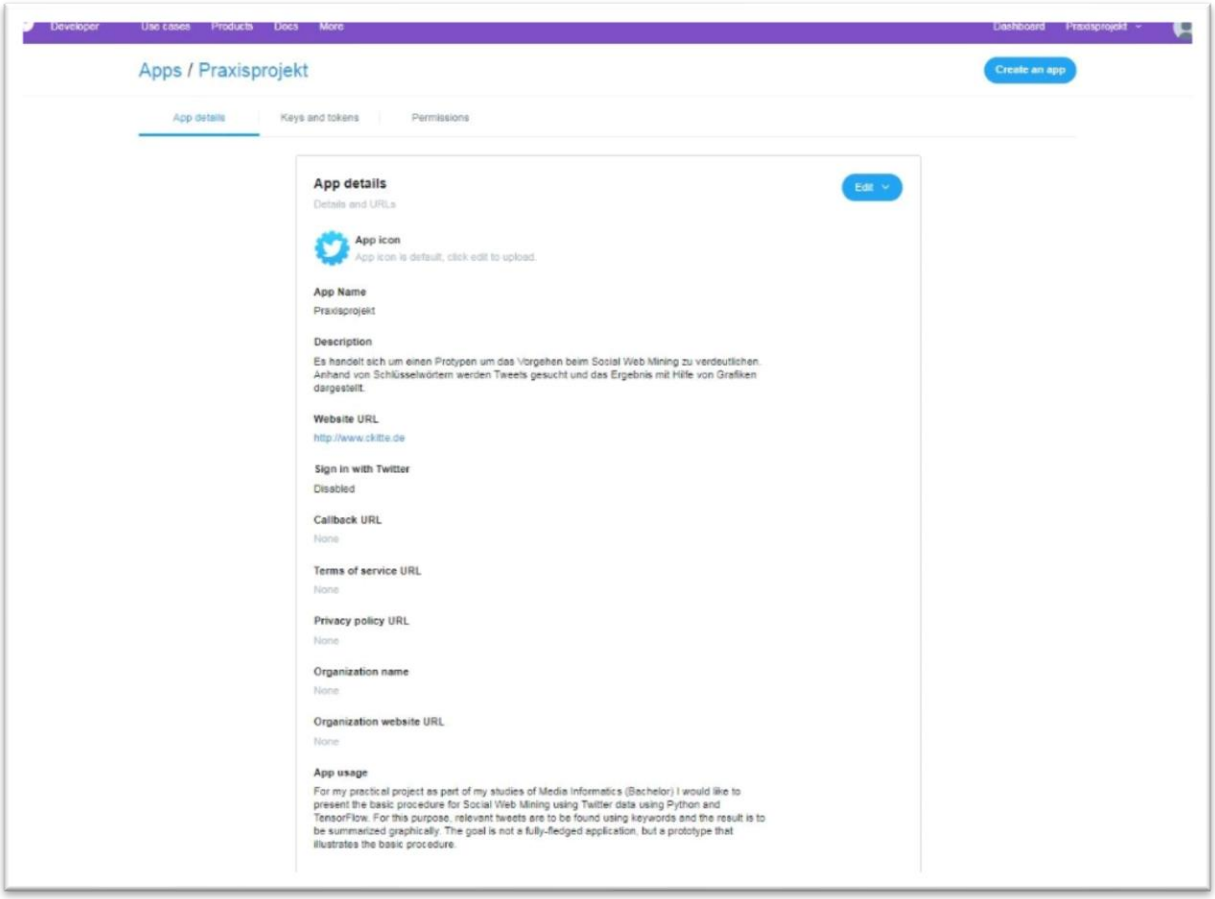



STATUS: IN PROGRESS
User profile
Account details
Use case details
Terms of service
Email verification
Why the questions?
We empower freedom of expression by providing a platform that protects the voices of our users — both on Twitter, and via our developer products. To help verify that all uses of Twitter data comply with our policies, we require additional information from developers signing up to use this service. Providing thorough answers will help us understand your use cases and will help expedite the evaluation of your application. Learn more about our restricted use cases.
Tell us about your project
What use case(s) are you interested in?
Select all that apply
Academic research
Advertising
Audience analysis
Chatbots and automation
Consumer / end-user experience
Engagement and customer service
Publish and curate Tweets
Student project / Learning to code
Topic analysis
Trend and event detection
Other
Describe in your own words what you are building
In English, please describe your product - the more detailed the response, the easier it is to review and approve. Be sure to answer the following:
What is the purpose of your product or service?
What will you deliver to your users/customers?
How do you intend to analyze Tweets, Twitter users, or their content?
How is Twitter data displayed to users of your end product or service (e.g. will Tweets and content be displayed at row level or in aggregate)?
For my practical project as part of my studies of Media Informatics (Bachelor) I would like to present the basic procedure for Social Web Mining using Twitter data using Python and TensorFlow. For this purpose, relevant tweets are to be found using keywords and the result is to be summarized graphically. The goal is not a fully-fledged application, but a prototype that illustrates the basic procedure.
Will your product, service, or analysis make Twitter content or derived information available to a government entity?
In general, schools, colleges, or universities do not fall under this category.
No
Yes
Continue


STATUS: IN PROGRESS
User profile
Account details
Use case details
Terms of service
Email verification
Read and agree to the Terms of Service
Scroll through to accept
Developer Agreement
Effective: May 25, 2018.
This Twitter Developer Agreement ("Agreement") is made between you (either an individual or an entity, referred to herein as "you") and Twitter, Inc. and Twitter International Company (collectively, "Twitter") and governs your access to and use of the Licensed Material (as defined below).Your use of Twitter's websites, SMS, APIs, email notifications, applications, buttons, embeds, ads, and our other covered services is governed by our general Terms of Service and Privacy Policy.
PLEASE READ THE TERMS AND CONDITIONS OF THIS AGREEMENT CAREFULLY, INCLUDING WITHOUT LIMITATION ANY LINKED TERMS AND CONDITIONS APPEARING OR REFERENCED BELOW, WHICH ARE HEREBY MADE PART OF THIS LICENSE AGREEMENT. BY USING THE LICENSED MATERIAL, YOU ARE AGREEING THAT YOU HAVE READ, AND THAT YOU AGREE TO COMPLY WITH AND TO BE BOUND BY THE TERMS AND CONDITIONS OF THIS AGREEMENT AND ALL APPLICABLE LAWS AND REGULATIONS IN THEIR ENTIRETY WITHOUT LIMITATION OR QUALIFICATION. IF YOU DO NOT AGREE TO BE BOUND BY THIS AGREEMENT, THEN YOU MAY NOT ACCESS OR OTHERWISE USE THE LICENSED MATERIAL. THIS AGREEMENT IS EFFECTIVE AS OF THE FIRST DATE THAT YOU USE THE LICENSED MATERIAL ("EFFECTIVE DATE").
IF YOU ARE AN INDIVIDUAL REPRESENTING AN ENTITY, YOU ACKNOWLEDGE THAT YOU HAVE THE APPROPRIATE AUTHORITY TO ACCEPT THIS AGREEMENT ON BEHALF OF SUCH ENTITY. YOU MAY NOT USE THE LICENSED MATERIAL AND MAY NOT ACCEPT THIS AGREEMENT IF YOU ARE NOT OF LEGAL AGE TO FORM A BINDING CONTRACT WITH TWITTER, OR YOU ARE BARRED FROM USING OR RECEIVING THE LICENSED MATERIAL UNDER APPLICABLE LAW.
I. Twitter API and Twitter Content
A. Definitions
1. Twitter Content – Tweets, Tweet IDs, Twitter end user profile information, Periscope Broadcasts, Broadcast IDs and any other data and information made available to you through the Twitter API or by any other means authorized by Twitter, and any copies and derivative works thereof.
2. Broadcast ID - A unique identification number generated for each Periscope Broadcast.
3. Developer Site – Twitter's developer site located at https://developer.twitter.com.
4. End Users – Users of your Services.
5. Licensed Material – A collective term for the Twitter API and Twitter Content.
6. Periscope Broadcast - A live or on-demand video stream that is publicly displayed on Twitter Services and is generated by a user via Twitter's Periscope Producer feature (as set forth at https://help.periscope.tv/customer/en/portal/articles/2600293).
7. Services – Your websites, applications and other offerings that display Twitter Content or otherwise use the Licensed Material as approved by Twitter through any onboarding process.
8. Tweet ID – A unique identification number generated for each Tweet.
9. Tweet – a short-form text and/or multimedia-based posting made on Twitter Services.
10. Direct Message - A text and/or multimedia-based posting that is privately sent on Twitter Services by one end user to one or more specific end user(s).
11. Twitter API – The Twitter Application Programming Interface ("API"), Software Development Kit ("SDK") and/or the related
By clicking on the box, You indicate that you have read and agree to this Developer Agreement and the Twitter Developer Policy, additionally as its relates to your display of any of the Content, the Display Requirements; as it relates to your use and display of the Twitter Marks, the Twitter Brand Assets and Guidelines; and as it relates to taking automated actions on your account, the Automation Rules. These documents are available in hardcopy upon request to Twitter.
Subscribe to our email list for product updates, developer news, and marketing communications.
Submit application


Developer
Email verification
Hi Christian Kitte!
Thanks for applying for a Twitter Developer account.
Please confirm your email address to complete your application.
Confirm your email
Thanks!
The Twitter Dev team
developer.twitter.com | @twitterdev
Twitter, Inc. 1355 Market Street, San Francisco, CA 94103


Developer
Use cases
Products
Docs
More
Dashboard
Apps
Create your first app
You'll need an app and API key in order to authenticate and integrate with most Twitter developer products. Create an app to get your API key.
Create an app



(2) Code zur Anreicherung von Tweets mit Text anhand deren IDs

Unter der Webadresse https://www.spinningbytes.com/resources/germansentiment/ kann eine Datei mit 9738 deutschen, klassifizierten Tweets heruntergeladen werden. Aus Gründen der Nutzungslizenz von Twitter dürfen keine größeren Mengen an Texten von Tweets allgemein zugänglich sein, weshalb in dieser Liste nur die IDs der jeweiligen Tweets zu finden sind.

Unten ist eine kurze Codesequenz zu sehen, welche die Abfrage unter Nutzung des Moduls Tweepy und entsprechender Zugangsdaten leistet. Da die Sequenz sehr einfach ist, wurde auf eine ausführliche Dokumentierung verzichtet.

Der Code kann einfach in eine Python Datei hineinkopiert und zur Ausführung gebrachte werden. Bei der Ausführung sind Lese- und Schreibrechte erforderlich.

```
import tweepy
import time
import sys

auth_helper = {'access_token': 'xxx',
               'access_token_secret': 'xxx',
               'consumer_key': 'xxx',
               'consumer_secret': 'xxx'}
```

```
auth = tweepy.OAuthHandler(auth_helper['consumer_key'], auth_helper['consumer_secret'])
auth.set_access_token(auth_helper['access_token'], auth_helper['access_token_secret'])

semantic_dict = {'negative': -1, 'neutral': 0, 'positive': 1, }

# https://stackoverflow.com/questions/16208206/confused-by-python-file-mode-w
result = open(r"c:\pfad\zur\Datei\mit\den\ids.csv", mode="a")

def call_twitter(id, sentiment):
    try:
        api = tweepy.API(auth)

        if (api != None):
            tweet = api.get_status(id)

            t = (str(id), tweet.text, str(semantic_dict[sentiment]), sentiment)
            t_string = ';'.join(t) + '\n'

            print(t_string)
            result.write(t_string)
    except tweepy.RateLimitError:
        time.sleep(15 * 60)
    except:
        print(sys.exc_info()[0], ' or not found')
        return

for line in open(
        r"c:\pfad\zur\Datei\mit\den\ids.txt"): # id, sentiment, …
        content = line.split()
        call_twitter(content[0], content[1]) # Stelle 0 und 1 enthalten die notwendigen Daten

        time.sleep(0.25) # zur Sicherheit um Zugriffsbeschränkungen zu umgehen
    else:
        print("Fertig")
        break
```

(3) Wiedergabe der originalen Texte und Urtexte der Testdaten

Urtexte im Sinne dieser Arbeit sind die Texte der ursprünglichen Tweets, auf denen eine Reaktion als einfacher Retweet oder kommentierter Quote Tweet erfolgte. Existiert kein Urtext, so erfolgte die Veröffentlichung nicht als Reaktion auf einen anderen Tweet.

Tweet 1:

Text: Vater fliegt Weihnachten bei Stewardess mit: Der Flugbegleiterinnenbegleiter https://t.co/8kTyCnMSBJ https://t.co/p1VdljsSmi

Tweet 2:

Text: RT @_sixtyniner_: Elftes Gebot:\n\nSchicke Weihnachten niemandem Gifs, den du das ganze Jahr ignoriert hast

Urtext: Elftes Gebot:\n\nSchicke Weihnachten niemandem Gifs, den du das ganze Jahr ignoriert hast

Tweet 3:

Text: RT @Saefken: Der \u00fcberwiegend islamische Irak hat Weihnachten auf Wunsch der christlichen Minderheit zum gesetzlichen Feiertag erkl\u00e4rt. Die\u2026,

Urtext: Der \u00fcberwiegend islamische Irak hat Weihnachten auf Wunsch der christlichen Minderheit zum gesetzlichen Feiertag erkl\u00e4rt. Die Weltreligion Islam ist offenbar pluraler und vielschichtiger, als die rechtsextremen Hassprediger es uns so oft glauben machen wollen. https://t.co/C6gM2oLSEz"

Urtext: Irak macht Weihnachten zum nationalen Feiertag https://t.co/7kBSEytbmn

Tweets 4:

Text: RT @Ich_bin_ich_666: #Weihnachten wurde in Somalia verboten:\n\n\"Wir sind ein muslimisches Land. Und es gibt null Toleranz f\u00fcr solche unislam\u2026

Urtext: #Weihnachten wurde in Somalia verboten:\n\n\"Wir sind ein muslimisches Land. Und es gibt null Toleranz f\u00fcr solche unislamischen Feiern in unserem Land.\"\n\nDas kann uns hier nicht passieren. Wir haben ja einen toleranten Islam. \nhttps://t.co/YY9Nn5lPlf

Tweet 5:

Text: RT @ohfamoos: Guten Morgen allerseits \ud83d\ude4b\ud83c\udf0d... wir hoffen ihr habt Weihnachten gut \u00fcberstanden \ud83d\ude00", "

Urtext: Guten Morgen allerseits \ud83d\ude4b\ud83c\udf0d... wir hoffen ihr habt Weihnachten gut \u00fcberstanden \ud83d\ude00",

Tweet 6:

Text: Alle Jahre wieder... Leider! Auch dieses Weihnachten mussten wir teils erheblich betrunkene Autofahrer aus dem Verkehr ziehen. So u.a. am Heiligen Abend auf der #A24 bei Neustadt- Glewe, als wir bei einem 38-J\u00e4hrigen einen Atemalkoholwert von 2,23 Promille feststellten. https://t.co/oo2NPrjEce

Tweet 7:

Text: Erstes Weihnachten ohne Jens? @SchiessKlaus wer ist Jens? Trauer herrscht f\u00fcr mich 400 Tote wieder durch einen Zunami ! Was soll diese unn\u00fctze Berichterstattung? Soviel Leid auf dieser Welt

Tweet 8:

Text: \u201eUnd das soll euch als Zeichen dienen: Ihr werdet ein Kind, das, in Windeln gewickelt,in einer Krippe liegt.\u201c\u2014 Lukas 2,12 #weihnachten\ud83c\udf84

#navidad\ud83c\udf84 #christmas #krippe #nacimiento #nativity #firstchristmasingermany #deutschland\ud83c\udde9\ud83c\uddea https://t.co/HHUH5gAV8p",

Tweet 9:

Text: RT @ConCrafter: Meine Eltern haben mir zu Weihnachten einen Alpaka-Wanderung geschenkt und es war das beste Geschenk aller Zeiten https://t\u2026

Urtext: Meine Eltern haben mir zu Weihnachten einen Alpaka-Wanderung geschenkt und es war das beste Geschenk aller Zeiten https://t.co/OB1SvpJQgi

Tweet 10:

Text: Bei @RTLde wissen sie auch nach #Weihnachten noch ganz genau, wie man seine Zielgruppe direkt anspricht... https://t.co/YBAMbSbHdh",