Konzeptpapier
Twitter-Dash

Hahn, Bastian Kleber, Martin Klier, Andreas Kreussel, Lukas Paris, Felix Ziegler, Andreas
b.hahn@oth-aw.de m.kleber2@oth-aw.de a.klier@oth-aw.de l.kreussel@oth-aw.de f.paris1@oth-aw.de a.ziegler1@oth-aw.de

***Zusammenfassung*—Ziel des Projekts ist es, ein Dashboard zu erstellen, welches Live-Informationen von der Twitter-API abruft und diese visualisiert.**

I. Einführung

Twitter ist eine der populärsten Informationsquellen im Internet. Auf dieser Seite werden täglich eine halbe Milliarde Kurznachrichten [4], sog. Tweets veröffentlicht. Eine solche Kurznachricht hat dabei eine Länge von maximal 280 Zeichen. Diese Tweets reichen von lustigen Bildern bis zur Verkündigung von politischen Entscheidungen, zum Beispiel des ehemaligen US-Präsidenten Donald Trump. Die schiere Masse an Daten macht es fast unmöglich einen Überblick zu behalten. Unter Zuhilfenahme von verschiedenen Maschine-Learning Verfahren können diese aber analysiert und interessante Informationen extrahiert werden, beispielsweise der zeitliche Verlauf mit welcher Intensität bestimmte Themen diskutiert wurden. In diesem Projekt soll ein Dashboard erstellt werden, welches die Tweets ausliest und Informationen zu diesen visualisiert. Das Projekt ist Teil der Vorlesung „Big Data und Cloud-basiertes Computing“ an der OTH Amberg-Weiden.

II. Verwandte Arbeiten

Es gibt bereits mehrere Systeme, welche sich mit der Analyse von Twitter-Daten beschäftigen. Diese Angebote lassen sich in drei unterschiedliche Gruppierungen einordnen:

Profil-Analyse: Bei der Profil-Analyse wird das Profil eines angegebenen Twitter-Accounts ausgewertet. Dabei werden vor allem persönliche Informationen und die Interaktionen mit anderen Accounts ausgewertet. Ein Beispiel für eine solche Plattform ist *tweetdeck* [2].

Trend Verlauf: Bei der Trend Verlauf Analyse wird der zeitliche Verlauf eines bestimmten Themas ausgewertet. Ein Beispiel für eine solche Plattform ist *trends24* [1].

Hashtag-Analyse: Bei der Hashtag-Analyse wird die Interaktion von Nutzern mit einem Themenfeld ausgewertet. Ein Beispiel für eine solche Plattform ist *trackmyhashtag* [3].

A. Vorgeschlagene Umsetzung

Unsere Umsetzung setzt eigene Schwerpunkte, welche in den Anforderungen definiert werden. Dabei werden die drei genannten Analysegebiete gestreift, wenn auch unterschiedlich stark.

III. Anforderungen

Anforderung 1

Als Twitter-Nutzer möchte ich die Hashtags sehen, die aktuell auf Twitter trenden, weil ich einen Überblick über das aktuelle Geschehen auf Twitter erhalten will.
Akzeptanzkriterien:

- Twitter-API Call
- Aufbereitung der Daten durch Backend
- Noch keine Persistenz
- Anzeige als Tabelle in einer Webseite

Anforderung 2

Als Twitter-Nutzer möchte ich den zeitlichen Verlauf über ein ausgewähltes Hashtag sehen, weil mich die Relevanz eines bestimmten Ereignisses interessiert.
Akzeptanzkriterien:

- Datenhaltung in Datenbank
- Periodische (mind. 15 min) Abfrage der Twitter-Daten durch Backend und Speicherung
- Angabe von Hashtag und Zeitrahmen
- Abfrage der Daten aus Datenbank
- Anzeige in Diagramm

Anforderung 3

Als Twitter-Nutzer möchte ich auch den zeitlichen Verlauf über mehrere ausgewählte Hashtags sehen, weil mich die Relevanz von mehreren Ereignissen interessiert.
Akzeptanzkriterien:

- Auswahl mehrere Hashtags
- Ablauf wie in III

Anforderung 4

Als Twitter-Nutzer möchte ich auch nicht durch Hashtags explizit definierte Topics erkennen können, weil ich einen Überblick über alle diskutierten Themen haben möchte.
Akzeptanzkriterien:

- Gruppierung aller Tweets der letzten Periode von der Twitter-API
- Analyse der Tweets mit Topic-Model
- Datenhaltung der gefundenen Topics in Datenbank
- Anzeige der Topics auf Webseite

Anforderung 5

Als Twitter-Nutzer möchte ich die aktuelle Stimmung zu einem Thema sehen, weil ich das Sentiment zu einem bestimmten Thema haben möchte.
Akzeptanzkriterien:

- Gruppierung der Tweets der letzten Zeitperiode nach Hashtag
- Analyse der Tweets mit Sentiment-Analyse
- Datenhaltung des Sentiments in Datenbank
- Anzeige des Sentiments auf Webseite

Anforderung 6

Als Twitter-Nutzer möchte ich einen Tweet angeben können um alle verwandten Informationen dazu sehen zu können. Als verwandte Informationen gelten:

- Zeitlicher Verlauf der im Tweet enthaltenen Hashtags
- Sentiment
- Topics

Akzeptanzkriterien:

- Eingabefeld für URL des Tweets
- Abfrage des Tweets über Twitter-API
- Sentiment-Analyse des Tweets
- Anzeige der verwandten Informationen

Anforderung 7

Als Twitter-Nutzer möchte ich sehen, welche Hashtags besonders oft mit einem ausgewähltem Hashtag zusammen vorkommen, weil mich der Zusammenhang mit anderen Themen interessiert.
Akzeptanzkriterien:

- Gruppierung der Tweets der letzten Periode nach Hashtag
- Zählen der Hashtags der gruppierten Tweets
- Datenhaltung der Hashtag-Beziehungspaare
- Eingabefeld für Hashtag
- Abfrage von Daten aus Datenbank
- Anzeige der verwandten Informationen

Anforderung 8

Als Twitter-Nutzer möchte ich Twitter-Benutzer abrufen können, um deren Metadaten angezeigt zu bekommen.
Als Metadaten gelten:

- Follower
- Tweets
- Engagement
- Activity

Akzeptanzkriterien:

- Eingabefeld für Twitter-Benutzer
- Abfrage der Nutzerdaten von Twitter-API
- Aufbereitung der Metadaten des Benutzers
- Anzeige der Metadaten des Benutzers

IV. Methoden

Datenakquise

Abfragen der Daten durch die offizielle API von Twitter oder durch Web-Scraper.

Datenfluss

Die Daten werden von einem oder mehreren Scraper- oder API-Containern erzeugt und von einem oder mehreren Backend-Containern verarbeitet. Diese Daten werden in einer Datenbank gespeichert, die Original Daten werden dabei nicht persistiert. Auf der Website werden die aus den Daten erhaltenen Informationen angezeigt. Der Datenfluss erfolgt dabei nach einem Push-Prinzip, das bedeutet, dass die Daten alle X Minuten (mind. 15 Min) neu von der API abgefragt und verarbeitet werden.

Backend

Als Backend werden mehrere Docker-Container verwendet, dabei werden für die Datenakquise .NET 6 Container verwendet. Die Verarbeitung der Daten erfolgt über Python-basierte Container, welche verschiedenste Python-Module und Funktionen verwenden. Die Kommunikation zwischen den Containern erfolgt über gRPC. Falls nötig kann ein Load-Balancer auf Basis von Kubernetes verwendet werden.

Datenbank

Zur Speicherung der Daten wird MongoDB oder Cassandra verwendet.

Frontend

Für die Interaktion mit dem Benutzer wird Angular, React oder Vue verwendet. Über eine REST-Schnittstelle oder gRPC-Schnittstelle des Backends kann auf benötigte Informationen zugegriffen werden.

Literatur

[1] trends24 [Online] https://trends24.in/germany/ (visited on 17. Mai 2022)
[2] tweetdeck [Online] https://tweetdeck.twitter.com/ (visited on 17. Mai 2022)
[3] trackmyhashtag [Online] https://www.trackmyhashtag.com/ (visited on 17. Mai 2022)
[4] [Online] https://www.brandwatch.com/de/blog/twitter-statistiken/#:~:text=Zahlen%20zur%20Twitter%2DNutzung,sind%206.000%20Tweets%20pro%20Sekunde. (visited on 17. Mai 2022)