Big Data Platform für Erdsystemdaten (Ophidia)

Nele Lips & Dominik Scherer

06.11.2017

Projekt „Big Data“

Betreuer: Jakob Lüttgau, Julian Kunkel

Universität Hamburg

Ziele

- Untersuchung der Leistungsfähigkeit von Ophidia
  - Geschwindigkeit
  - Funktionsumfang
  - Usability
- Feststellen, ob Ophidia Mehrwert bringt
- “Gebrauchsanleitung” ausarbeiten

Was ist Ophidia?

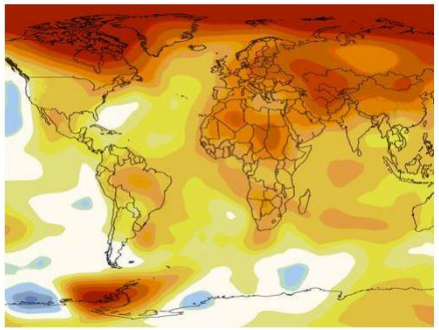
- Big Data Analytics Framework
- Entwickelt am Euro Mediterranean Centre on Climate Change (CMCC), erster Release 1.2.2016
- Support Programm beim Umgang mit Big Data in der Wissenschaft (erste Tests mit Klimadaten)

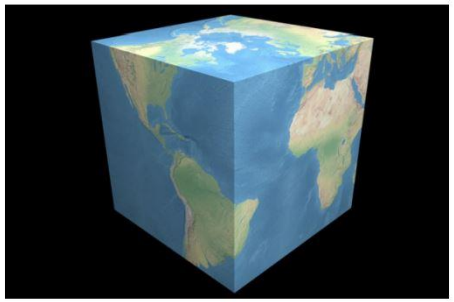
Ophidia Architektur

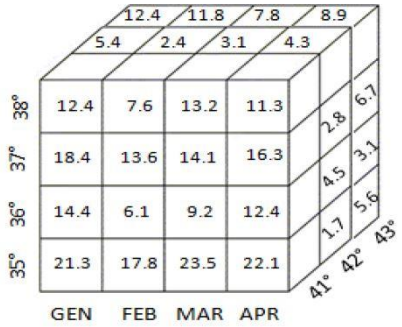
- Bietet Zugriffs-, Analyse und Mining Funktionalitäten
- Verwendet
  - Wissenschaftliche numerische Bibliotheken
  - Verteiltes und hierarchisches Speichermodell
  - Paralleles Software Framework basierend auf MPI
- Erweitert relationale Datenbanksysteme um
  - Primitive
  - Datentypen

Ophidia Architektur

- Array Based Data (multidimensional)







Quelle: Ophidia: a big data analytics framework, Dr. Sandro Fiore, [1]

Ophidia Architektur

- Aufbau
  - Server front-end
  - OphidiaDB
  - Compute nodes
  - I/O nodes
  - Storage system

Zeitplan

- Installation von Ophidia & Hintergrundwissen
  - Oktober
- Einarbeitung/Testen von Demobeispielen
  - November
- Benchmarken mit realen Testdaten (vom DKRZ)
  - Dezember
- SQL
  - Januar

Quellen:

- [1] http://aims-group.github.io/pdf/f2f2014/ophidia.pdf
- S. Fiore, A. D’Anca, C. Palazzo, I. Foster, D.N. Williams, G. Aloisio, Ophidia: Toward Big Data Analytics for eScience, In Procedia Computer Science, Volume 18, 2013, Pages 2376-2385
- https://github.com/OphidiaBigData/ophidia-analytics-framework

AI for a computer game

Gaming AI in der Spring RTS Engine

Valentin Krön und Friedrich Braun

NOTA

(Not Original Total Annihilation)

- Kostenfreie Engine für RTS
- Eins von vielen Möglichen Spielen für Die Spring RTS Engine
- Dem spiel „Total Annhilation“ nachempfunden
- Verschiedene Einheiten Arten, Ressourcen Management und Einheiten Micro
- KI soll das spielen

Die KI

- Auf Basis von Existierenden Bots einen Eigenen entwickeln
- Der Bot soll in Lage seine Entscheidungen während des Spiels selbst zu treffen
- Wir lassen den Bot gegen andere existierende Bots antreten
- Anschließen wird die KI durch Evolutionäre Algorithmen weiterentwickelt

Tools

- Als Programmiersprache wird C++ verwendet
- Dazu das Deep Learning Framework „Caffee“ developed by Berkeley AI Research
- Spring RTS selbst ist eine Engine und bietet auch tools für das Entwickeln von KI‘s an (C++ Wrapper API)
- Tool zur Auswertung der Game Logs

Zeitplan

- Bis Anfang Dezember: der Bot soll lauffähig sein
- Bis Ende Dezember: die Auswertung der Logs soll stehen (inkl. Auswertungsfunktion) und der Bot soll in der Lage sein reaktive Entscheidungen zu treffen
- Bis Ende Januar: der evolutionärer Algorithmus soll stehen (d.h. es soll dann möglich sein das Training zu beginnen)
- während Februar: trainieren der KI

Wolkenkamera: Big Data Projekt
Themenvorstellung

Marcel Steger, Jan Zickermann

Universität Hamburg

6. November 2017

Übersicht

1. Projektthema
2. Preprocessing
   - Entkrümmung
3. Supervised Learning
4. Unsupervised Learning
5. Zeitplan

Projektthema: Wolkenkamera

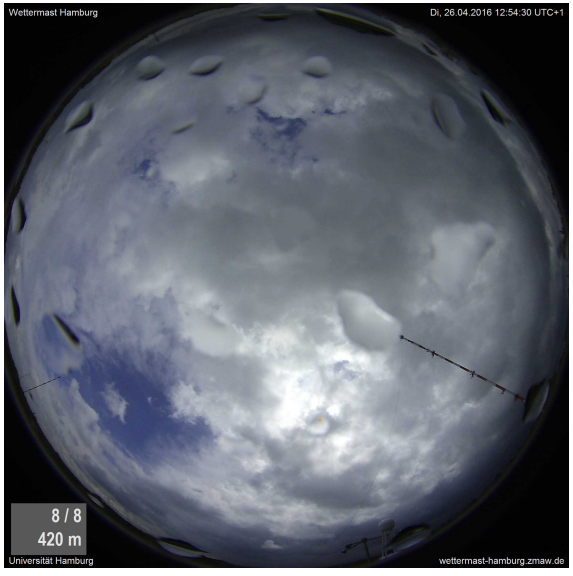
Ziel: anhand von Photographien einer stationären Kamera den Bewölkungsgrad zu ermitteln. Hierfür sollen neuronale Netze verwendet werden.

[1]

python: tensorflow (machine learning), scipy (preprocessing)

Preprocessing

- Aussortierung der 'noisy labels'



[2]

- Erkennung der Sonne und Differenzierung von Wolken
- Bestimmung des Messpunktes des Ceilometer

Entkrümmung

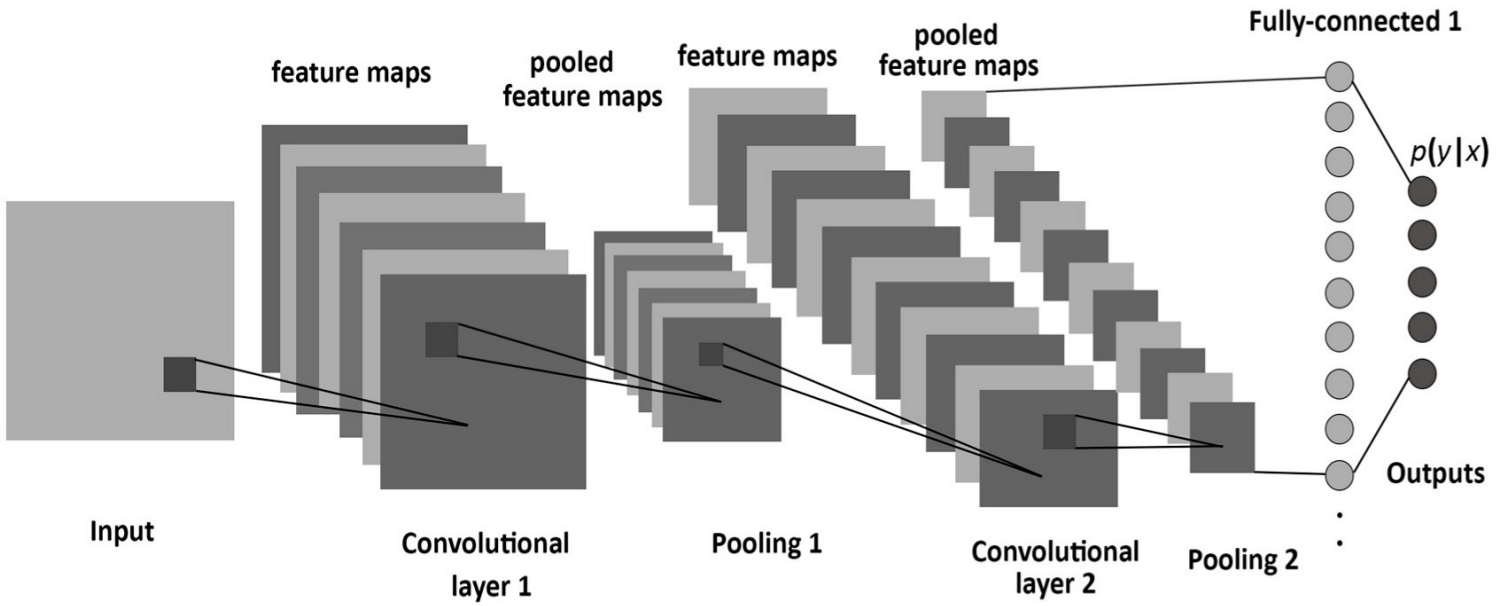
- $\cos\theta \rightarrow$ verzerrt an den Rändern

Supervised Learning

- verfügbare Labels:
  - rain (Regensensor)
  - height (Ceilometer)
- Klassifizierung von Wolkenbild nach Regen
- preprocessed data $\rightarrow$ *Convolutional Neural Network*

Convolutional Neural Networks

- lokale Informationen verwenden, wie das Auge
- *convolution*: Filter die über Pixel fahren
- *pooling*: Pixel reduzieren (z.B. 2x2 max-pooling)
- Wolkenbild $\rightarrow$ Regenwahrscheinlichkeit



[3]

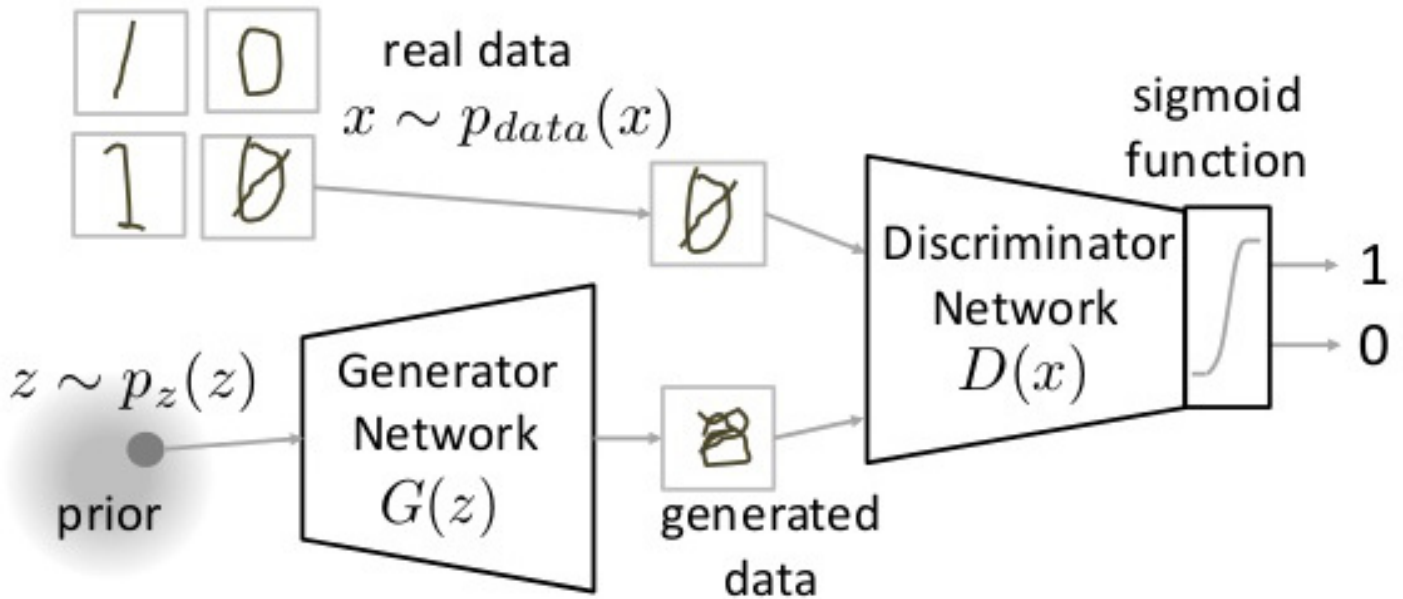
Unsupervised Learning (GAN)

- Ziel: Wolkenbilder zu generieren basierend auf vorangegangenen Bildern
- *generator*: realistisches Bild (nach Urteil von *discriminator*)
  - wird trainiert durch output von *discriminator*
- *discriminator*: entscheidet real/fake
  - wird trainiert mit *real data* input

Generative Adversarial Networks

$$\min_G \max_D V(D,G)$$

$$V(D,G) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z))]$$



[4]

GAN Probleme

- Folgebilder generieren:
  - Informationen der vorangegangenen Bilder erhalten
  - möglicherweise *zu wenige Informationen* für Generator
- Performance:
  - Bildgröße muss reduziert werden
  - reduzieren von neurons/layers $\rightarrow$ Tests notwendig

Zeitplan

1. supervised learning (Regen) für CNN Model
   - bis Dezember
2. CNN Model für GAN (unsupervised learning)
   - Januar
3. training / finetuning

Bildquellen:

- Wolkenbild, 04.11.2017 : https://github.com/LEX2016WoKaGru/pyclamster/blob/master/examples/images/wettermast/Image_Wkm_Aktuell_2.jpg?raw=true
- Wolkenbild mit Regen, 05.11.2017 : https://github.com/LEX2016WoKaGru/pyclamster/blob/master/examples/images/wettermast/Image_Wkm_Aktuell_1.jpg?raw=true
- CNN, 05.11.2017: http://www.mdpi.com/entropy/entropy-19-00242/article_deploy/html/images/entropy-19-00242-g001.png
- GAN, 04.11.2017: https://image.slidesharecdn.com/generativeadversarialnetworks-161121164827/95/generative-adversarial-networks-11-638.jpg?cb=1480242452

Semantische Suche mit Apache Solar

Eike Nils Knopp, Minh Hieu Nguyen

Themenstellung

In wie weit kann Apache Solr für semantische Suchen (ähnliche Begriffe, Synomyme) eingesetzt werden?

Konkretes Ziel

Erstellung einer Web-App

- › Intuitive semantische Suche
- › Passend für komplexe wissentschaftliche Daten

REST-Anbindung der Web-App an Apache Solr

Features

Basic Website

- Suchfeld
- Ergebnisliste
- Kategorien / Filter

Features

- Statistics (User query statistics)
- Machine Learning „Learning to Rank“
- Intuitives durchsuchen spezieller Felder
  - *project_name*
  - *entry_type*

Langzeitziele

- Tags und Kategorien vereinheitlichen (CleanUp)
  - › Unterschiedliche Schreibweisen der Selben Sachen zusammenführen
- Intelligente Felder
- Stemming der Suchbegriffe
- Suche nach Synonymen des Suchbegriffs
- Result Clustering mit Carrot2

Vorgehen

Frontend:

- › ReactJS

Backend:

- › Noch nicht ganz absehbar, ob benötigt
- › Falls benötigt: NodeJS

Apache Solr

- › Version 7.1.0

Vorgehen

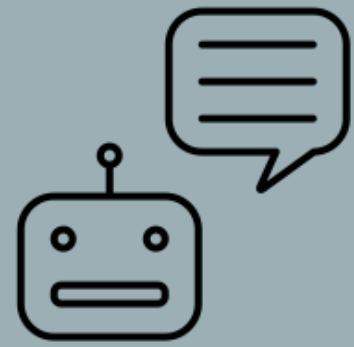
Erstellung einer rudimentären Website in ReactJS (Testumgebung)

Anbindung von Apache Solr an die Website

- › Anbindung von Filtern etc.
- › Einbindung von Plugins wie User Query Statistics

Chat bot

Paul Offner
Merlin Sewina
Felicitas Castrian



ZIEL

Implementierung eines Chat bots, welcher auf Level1 sowie Level 2 Supportanfragen von Usern über eine Webschnittstelle antworten und auf die entsprechenden Links verweisen kann.

HERANGEHENSWEISE

- 1. Crawler
- 2. Programm zur Textanalyse
- 3. Anforderungsanalyse an Chat bot
- 4. Chat bot
- 5. Web Schnittstelle für User Anfragen/ Antworten

1. CRAWLER

- Crawler für DKRZ Seite
- Speichern aller gefundenen Html-Links mit Inhalt in Dateien in einem Verzeichnis.
- Reduzierung der Dateien auf die Textelemente.

2. TEXTANALYSE

- Programm zur Analyse der abgespeicherten Dateien.
- Auffinden von Schlagwörtern
- Verweisung vom Chat bot auf die entsprechenden Links bei Anfrage

→Verschiedene Ansätze zur Analyse von Texten.

3. ANFORDERUNGSANALYSE CHAT BOT

- Analyse des Umfangs und der Art möglicher Supportanfragen an unseren Chat bot.

→ Detailliertere Auseinandersetzung mit Website des DKRZs.

4. CHAT BOT

- Implementierung Chat bot auf Basis der Anforderungsanalyse.

5. WEB SCHNITTSTELLE

- Web Schnittstelle (REST API) für User Anfragen/ Antworten an Chat bot über HTTP.

ZEITPLAN

30.10-05.11	Crawler implementiert
06.11-12.11	Crawler fertigstellen / Textanalyse Ideen
13.11-29.11	Textanalyse umsetzten
20.11-26.11	Textanalyse
27.11-03.12	Anforderungsanalyse Support
04.12-10.12	Chat bot / Zwischenpräsentation
11.12-17.12	Chat bot
18.12-22.12	Chat bot
08.01-14.01	Webschnittstelle / Zwischenpräsentation
15.01-21.01	Webschnittstelle
22.01-28.01	Projektbericht / Präsentation
29.01-03.02	Projektbericht fertigstellen

BILDQUELLE

- https://d30y9cdsu7xlg0.cloudfront.net/png/852157-200.png

Analyse von News-Artikeln – Sentiment Analysis mit word2vec

Maike Schubert (5mschube@informatik.uni-hamburg.de)
Raffael Diestel (5diestel@informatik.uni-hamburg.de)

Detaillierte Themenvorstellung

- Analyse von News
- Sammlung großer Daten Sets von Artikeln
  - beschränken auf bestimmte Topics (bspw. Politik, Business)
- Sentiment-Analysis mit word2vec
  - Allgemein auf Herausgeber bezogen und themenbezogen

Inwiefern lässt sich durch Sentiment-Analysis mittels word2vec das allgemeine Sentiment sowie auch die Position zu bestimmten Themen von den verschiedenen News-Portalen ermitteln?

Ansatz/Methodik

- Vorgehen:
- Sammlung von Daten mittels Crawler → Data Set erhalten
- Extrahieren von Text und Schlüsselinformationen
- Analyse und Visualisierung der Daten mit word2vec
- Auswertung der Ergebnisse

- Themen:
- Natural Language Processing
- Machine Learning
- Text Analysis
- Data Visualization

Zeitplan

- Daten sammeln
  - Dauer: ca. 2 Monate
- relevante Informationen extrahieren
- Daten analysieren/visualisieren
- Auswertung der Ergebnisse

Exploration of News

Tatyana Galitskaya, Sara Yüksel, Alexander Spikofsky

Ziele

- Unterschiede und Gemeinsamkeiten zwischen amerikanischen, britischen und deutschen News herausfinden
- Fokus auf internationale Nachrichten in englischer Sprache
- Zu untersuchende Seiten u.a. BBC, The New York Times, Spiegel Online, …

Methoden & Tools

- Ideen:
  - Themenidentifikation
  - Ähnlichkeiten zwischen Artikeln
  - Association Rule Mining, Wort-Kontexte
  - Reposting-Analyse, Sentiment-Analyse
  - Größe der Rubriken
  - Textlänge pro Rubrik
  - Anzahl Artikel pro Zeiteinheit
- Analysen durch Keywords, Bag of Words, POS, N-Grams, …
- Tools: Pandas, scikit, Jupyter, ggplot2, Gephi, Tableau

Zeitplanung

- Aufsetzen des Crawlers
  - RSS Feed Liste erstellen (bis 06.11.17)
  - BeautifulSoup einrichten zur Extraktion des Reintextes (bis 06.11.17)
  - Crawler laufen lassen (über gesamte Projektzeit)
- Text Mining
  - Analysemethoden auswählen und einarbeiten
  - Anwendung der Methoden auf gesammelte Daten (ab 06.11.17 über gesamte Projektzeit)
- Ergebnisvisualisierung
  - (Laufend sobald Daten aussagekräftig analysiert wurden)
- Erweiterung um weitere Aufgaben
  - (falls am Ende noch Zeit ist)

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

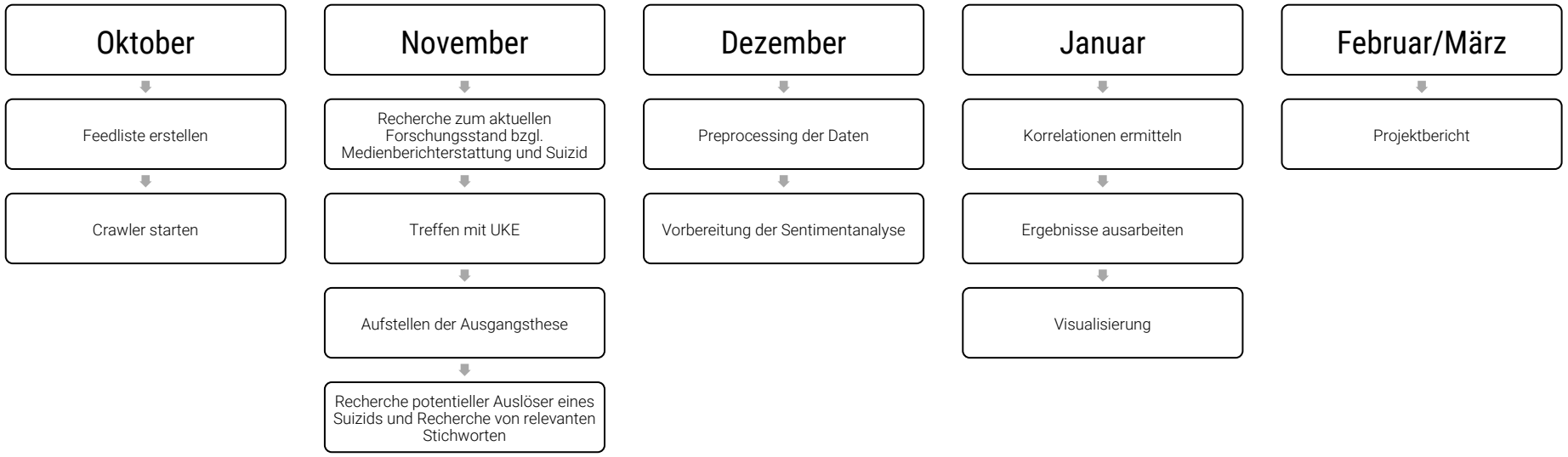
Nina Arndt, Melanie Budde, Ariana Sliwa

Analyse von News zur Suizidprävention

Thema // Projektziel

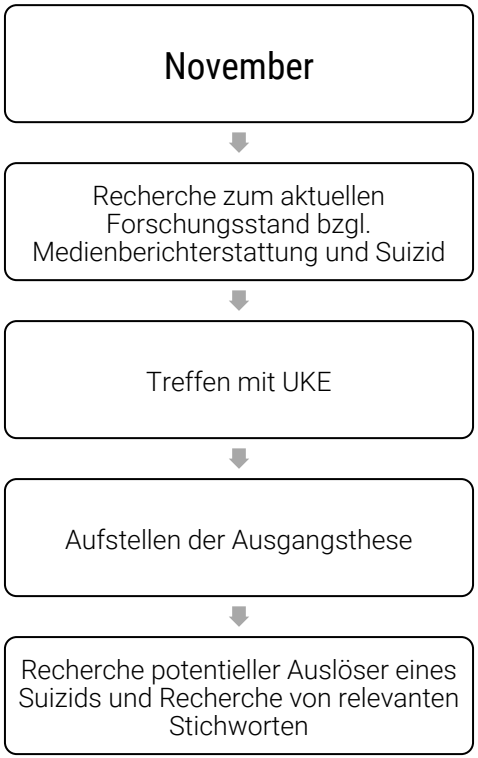
Kann ein Werther-Effekt nachgewiesen werden? Besteht ein Zusammenhang zwischen der (Online-) Medienberichterstattung (der Metropolregion Hamburg) und der lokalen Suizidrate?

Zeitplan und Meilensteine

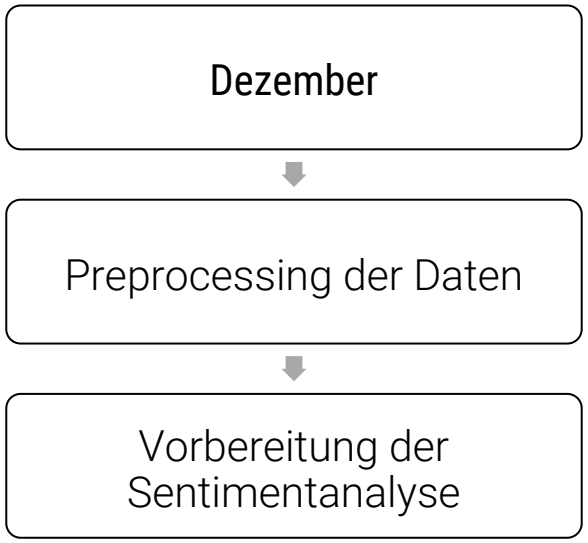


Zeitplan und Meilensteine

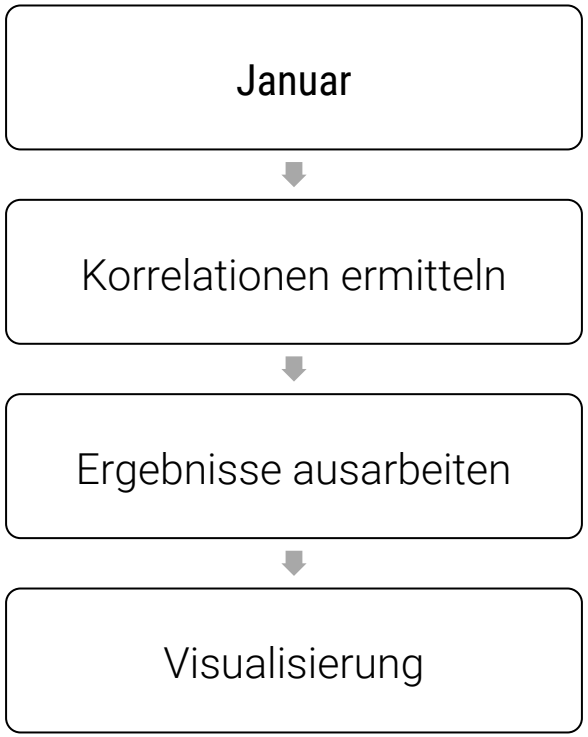
Zeitplan und Meilensteine



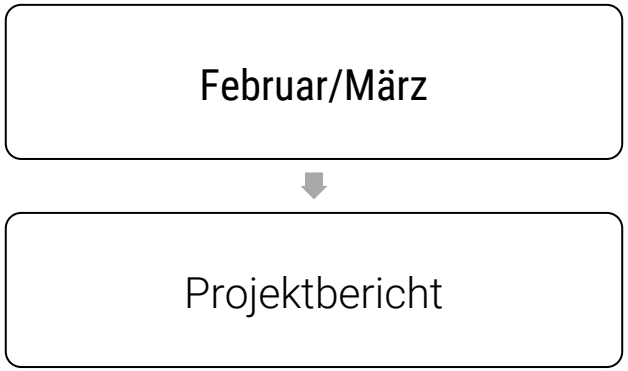
Zeitplan und Meilensteine



Zeitplan und Meilensteine



Zeitplan und Meilensteine



Fragen

- Können wir vergangene Artikel mit einbeziehen?
- Gibt es eine Möglichkeit, Twitter-Posts nach Hashtags zu crawlen?
- Wie detailliert sind die statistischen Daten des UKE? Geschlecht, Alter, Zeitpunkt, Art, etc. -> Tbd. bis zum Termin
- Analyse mittels welcher Tools? R? word2vec?

Feedback // Vorschläge