

MYCO GRUPPE 3

Cloud Computing Technology

Prof. Dr.-Ing. Peter Thies,
Prof. Dr.-Ing. Christoph Kunz

Datum: 30.01.2020

Christoph Zeltwanger, Georg Erich ,Lisa Kryszewski, Michael Schulz,
Philipp Stransky



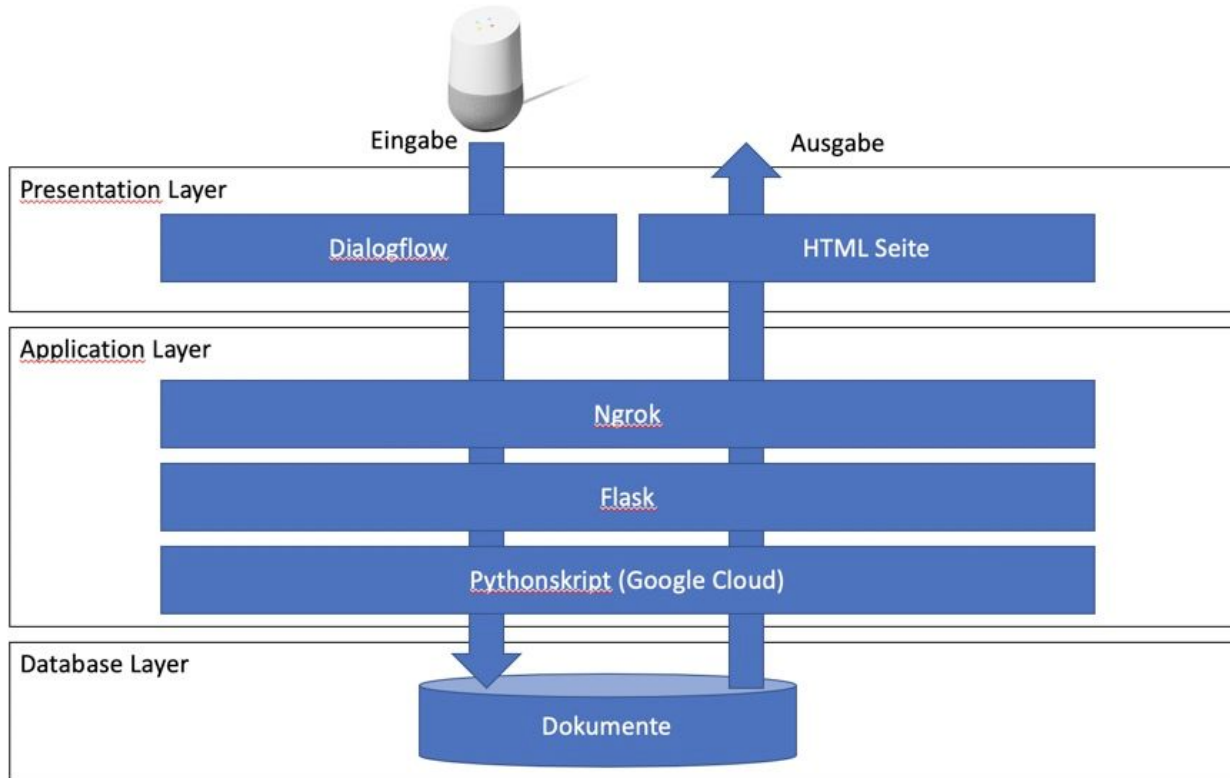


AGENDA

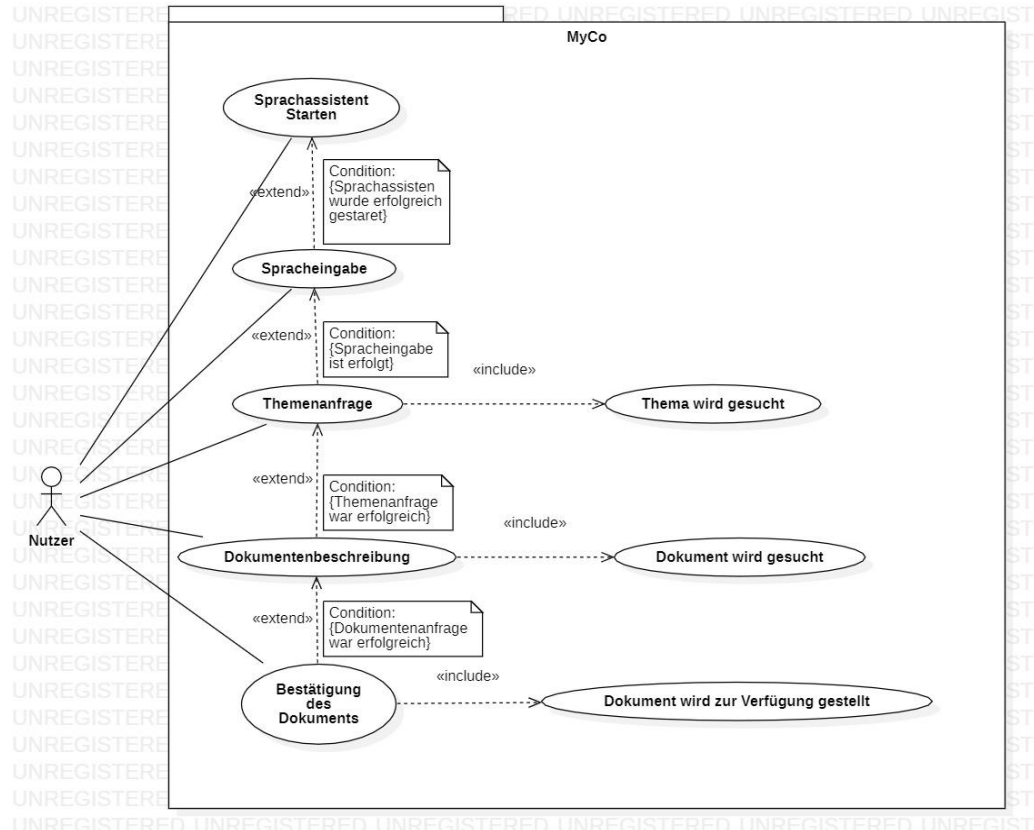
- 1.0 Erläuterung unseres Zielsystems
- 2.0 Use Case Sprachassistent
- 3.0 Klassendiagramm Gesprächskontext
- 4.0 Erzeugung der Trainingsdaten
- 5.0 Training des Netzes
- 6.0 Demonstration



1.0 Erläuterung unseres Zielsystems

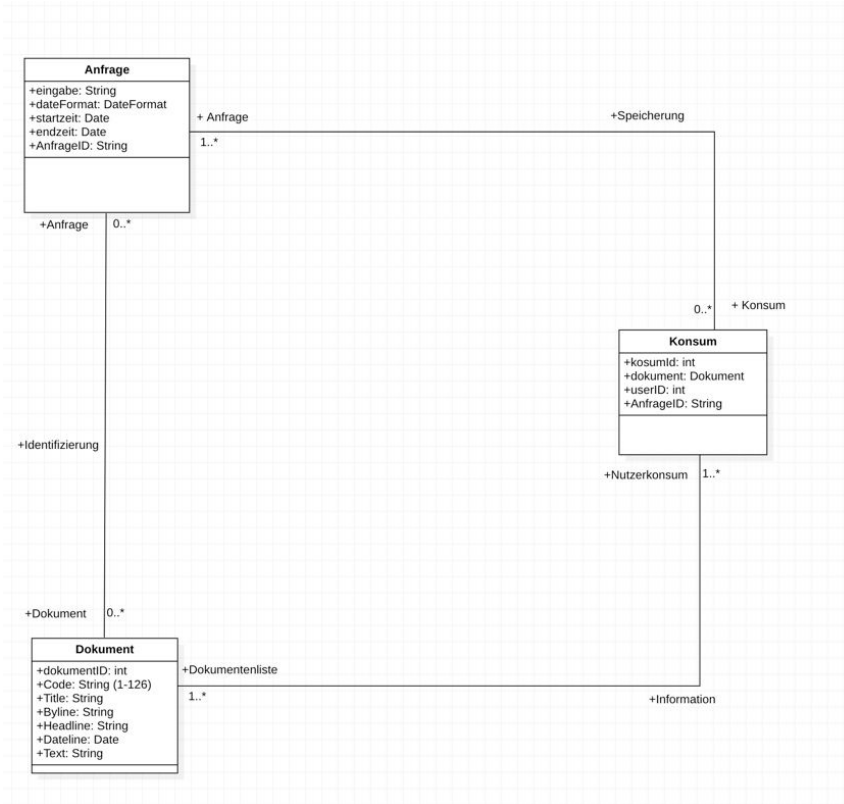


2.0 Use Case Sprachassistent





3.0 Klassendiagramm Gesprächskontext



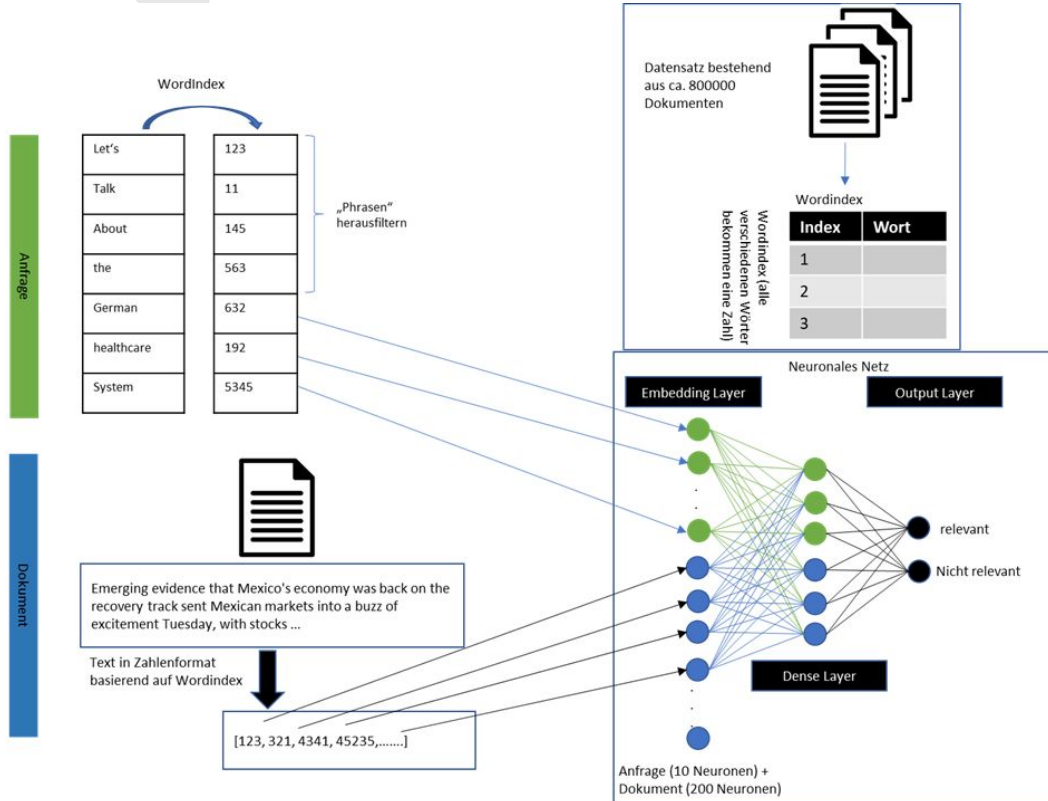
4.0 Erzeugung der Trainingsdaten

```
2.py
Themensuche.py  Themensuche 2.py  Test.py  Themensuche 2
1 import xml.etree.ElementTree as ET
2 import glob
3 import os
4
5 from nltk.tokenize import word_tokenize
6
7
8 # Hier das Verzeichnis angeben, indem nur noch die 4 verschiedenen Ordner drinnen liegen
9 dir = '/Users/philippstransky/Documents/HDM/Cloud/RC1 subset'
10 i = 0
11
12 for file in glob.glob(os.path.join(dir, '*/*.xml')):
13     with open(file) as f:
14
15         tree = ET.parse(file)
16         root = tree.getroot()
17         text = ""
18
19         for node in tree.iter('text'):
20             for elem in node.iter():
21                 if not elem.tag == node.tag:
22                     text = text + elem.text
23
24
25         tokenizedtext = word_tokenize(text)
26
27         ###Beispiel für ein Wort das gesucht werden muss, jeweils die nichtverwendete Art mit # an jedem Zeilenanfang auskommentieren
28         if 'car' in tokenizedtext:
29             print('file: ' + file)
30             print(text)
31             print('_____')
32             i = i + 1
33         else:
34             i = i + 1
35
36         ###Beispiel für zwei Worte die gesucht werden sollen, jeweils die nichtverwendete Art mit # an jedem Zeilenanfang auskommentieren
37
38         #.....if 'politics' in tokenizedtext:
39         #.....if 'EU' in tokenizedtext:
40         #.....print('file: ' + file)
41         #.....print(text)
42         #.....print('_____')
43         #.....
```

- Reduktion Datensatzes auf ca. 12.000 Dokumente
- durchsuchen reduzierter Datensatz mit Python Skript nach Themen
- Notation Kategorie mit 10 relevanten Dokumenten und 10 nicht relevanten Dokumenten
- Notation verschiedenster Trainingsdaten zum trainieren des Netzes



5.0 Training des Netzes



- Dokumente mit Wordindex umwandeln
- Input für neuronales Netz
 - Trainingsdaten + Dokumente
- Output
 - relevant
 - nicht relevant
- relevante Dokumente werden in einer Liste ausgegeben





6.0 Demonstration

