VERSICHERUNGS KAMMER

Coding Dojo – KI Freitextanalyse

Natural Language Processing

14. Oktober 2021

VERSICHERUNGS KAMMER BAYERN · SAARLAND Versicherungen · FEUERSOZIETÄT · UKV · URV · BavariaDirekt

Finanzgruppe

Intro

Definition der Daten

Meine erste Aufgabe - Korpusanalyse

Meine zweite Aufgabe - Schlagworte

VERSICHERUNGS KAMMER

Intro | Coding Dojo

Unser gemeinsames Problem

„Es liegt eine Textdatei am Laufwerk und wir wissen nur, dass es sich um Klappentexte von Büchern handelt, aber erstmal nicht mehr. Wir wissen nicht was die wichtigen Wörter pro Klappentext sind und was diese in Genre unterscheiden lässt.“

„Im Coding Dojo nehmen wir uns die Datei genauer vor und zwingen diese Datei anhand von Verfahren aus dem Bereich Computerlinguistik und Natural Language Processing mehr von sich preis zu geben.“

- Übersicht über Daten die innerhalb von Textdokumenten stecken (Text hat keine große Datenvielfalt auf den ersten Blick)
- Vorbereitung der Textdaten und Extraktion/Berechnung statistischer Merkmale

Keine Musterlösung

Es gibt nur gute Diskussionen

Definition der Daten

Original Daten von einer Kaggle Challenge (Randomhouse Verlag)

Ergänzende Informationen zur Beschreibung der Daten

Tag	Beschreibung
<book> </book>	Buchinformationen
<title> </title>	Buchtitel
<body> </body>	Klappentext
<copyright> </copyright>	Copyright Informationen
<categories> </categories>	Beinhaltet alle Kategorien eines Buches
<category> </category>	Beinhaltet das tatsächliche Genre und übergeordnete Genre (Kategorien-Hierarchie)
<topic d="n"></topic>	Name des Themas bzw. Kategorie eines Buches je nach Ebene der Kategorien-Hierarchie. Die spezifischste Kategorie wird markiert durch label = True.
<author> </author>	Autor des Buches
<published> </published>	Veröffentlichungsdatum
<isbn> </isbn>	ISBN
<url> </url>	Quelle der Buchinformationen

Definition der Daten

Meine Vorbereitung für das Dojo:

https://github.com/MaxNLP/codingdojo.git

- Extraktion des Klappentextes und des Hauptgenre
- Erzeugung eines csv mit „#“ als Seperator
- ISBN#Titel#Genre#Klappentext
- Für Implementierung und Debugging habe ich eine debug.csv erzeugt

DAS Korpus = **Buchsammlung-Klappentexte (B)**

Meine erste Aufgabe – Korpusanalyse | Frequenzlisten

Frequenzliste(Wörter zählen) von Token unabhängig von Genre und Ausgabe der Top n-Wörter (n=[10,20,50])

Einlesen:

1. Verwendung des Klappentextes
2. Einlesen der Daten und Verarbeitung während des lesens
3. Erzeugung einer Datenstruktur Wort:Counter

Tokenisierung:

1. Entfernen der Punktuation (über Liste von Punktuationszeichen wie !,.?! …)
2. Entfernen von Leerzeichen

Normalisierung:

1. Entfernen von Stoppwörtern (z.B.: und, also, als, …)
2. Entfernen von Zahlen
3. Ändern des Casing (alles mit lowercase – Nachteile vs. Vorteile ?)

Meine erste Aufgabe – Korpusanalyse | Frequenzlisten

Frequenzliste(Wörter zählen) von Token abhängig von Genre und Ausgabe von den Top n-Wörtern (n=10) pro Genre.

Einlesen:

1. Verwendung des Klappentextes
2. Einlesen der Daten und Verarbeitung während des lesens
3. Erzeugung einer Datenstruktur (Tipp: Für jedes Genre ein Dictionary, Word:Counter)

Tokenisierung:

1. Entfernen der Punktuation (über Liste von Punktuationszeichen wie !,.?! …)
2. Entfernen von Leerzeichen

Normalisierung:

1. Entfernen von Stoppwörtern (z.B.: und, also, als, …)
2. Entfernen von Zahlen
3. Ändern des Casing (alles mit lowercase – Nachteile vs. Vorteile ?)

Meine erste Aufgabe – Korpusanalyse | Frequenzlisten

Frequenzliste(Wörter zählen) von Token abhängig von Buch und Ausgabe von den Top n-Wörtern (n=5) pro Buch.

Einlesen:

1. Verwendung des Klappentextes
2. Einlesen der Daten und Verarbeitung während des lesens
3. Erzeugung einer Datenstruktur (Tipp: Für jedes Buch ein Dictionary, Word:Counter)

Tokenisierung:

1. Entfernen der Punktuation (über Liste von Punktuationszeichen wie !,.?! …)
2. Entfernen von Leerzeichen

Normalisierung:

1. Entfernen von Stoppwörtern (z.B.: und, also, als, …)
2. Entfernen von Zahlen
3. Ändern des Casing (alles mit lowercase – Nachteile vs. Vorteile ?)

Meine zweite Aufgabe – „Simple“ Schlagworte

- Berechnung des Seltenheitsmaßes TF-IDF der Wörter pro Buch
- Welche Wörter sind am aussagekräftigsten in Bezug auf das Seltenheitsmaß pro Buch ?

Vorgehen:

1. Berechnung TF (Term-Frequency) -> Term (hier:Token) pro Buch
2. Berechnung DF:

   |B| Gesamtzahl der Bücher in Buchsammlung (Korpus)

   DF(t, B) Anzahl der Bücher unserer Buchsammlung B in denen der Term vorkommt

3. Berechnung IDF (Inverse Document Frequency):

   IDF(t,B) = log{ (|B| + 1) / (DF(t,B) + 1) }

4. Ausgabe von Token, DokumentId, TF, DF, IDF, mult(TF,IDF)