Freie Universität Berlin

Prof. Dr. Agnès Voisard, Nicolas Lehmann

Datenbank Systeme, SoeSe 2017

Project 2.Iteration

TutorIn: Hoffman Christian
Tutorium 3, Gruppe 22

Ingrid Tchilibou, Emil Milanov, Boyan Hristov

1. Juni 2017

Allgemein

Link zum Projekt: https://github.com/gancia-kiss/dbs_projekt

1.Aufgabe: Datenbankschema erstellen

Link zum *.sql* Datei:
https://github.com/gancia-kiss/dbs_projekt/blob/master/DatabaseDump.sql

Die letzte Iteration hatten wir ein paar 'Count' Attributen addiert, wir haben aber uns später entschieden, dass wir sie nicht brauchen. Diese Iteration, aber wir haben bemerkt, dass wir eigenltich diese Attribute brauchen.

Zusätzlich haben wir auch ein Count Attribut, damit wir aufzählen können, welche Paar von Hashtags am häufigsten vorkommt.

Wir haben uns entschieden selber IDs von Tweets zu erstellten und zwar mit Python UUID. Das Ergebnis kann größere Integer sein, deswegen haben wir als TWEET.ID 'bigint' als Datentyp benutzt.

Bei uns ist der Inhalt der Hashtags die primäre Schlüssel. Wir haben auch als Erinerrung der Attribut 'textlowercase' genannt, um zu wissen, nur kleine Buchstaben zu verwenden.

```
pg_dump election > DatabaseDump.sql      # Dump Datei erstellen

psql election < DatabaseDump.sql         # Datenbank vom .sql Datei importieren
```

2.Aufgabe: Datenbereinigung

Link zum Datenreinigungsprogramm:
https://github.com/gancia-kiss/dbs_projekt/blob/master/programs/cleaner.py

Nach tiefere Datenanalyse haben wir zwei Hauptprobleme identifiziert:

1. Manche Tweets sind abgeschnitten und ein Teil von denen sind 'Truncated'. Wir wussten aber nicht, wie wir schnell die Tweets rekonstruiren können. Deswegen haben wir die gezählt und haben festgestellt, dass nur 2% der Tweets abgeschnitten sind. Es wäre für uns am leichtesten, alle solchen Tweets einfach zu löschen.

Wir haben bemerkt, dass alle abgeschnittenen Tweets auf '...' dann einen Link endeten. Es ist aber herausgekommen, dass '...' in solchen Tweets nur einen Unicode Zeichen war. Deswegen müssten wir im Python Program überprüfen, ob der Zeichen $u'\backslash u2026'$ im Körper des Tweets vorkommt.

2. Die vorgegebene Einstellungen der Libre Office Calculator hat versucht mit Codierung 'UTF-8' geöffnet. Dann gab es aber Probleme mit manchen Symbolen. Apostrophe, manche Sonderzeichen und vermutlich auch Emojis wurden nicht angezeigt. Nach kurze Analyse haben wir festgestellt, dass wenn wir die Tabelle mit Codierung 'Windows 1252' öffnen, werden Apostrophen und wichtige Sonderzeichen normal dargestellt, und Emojis und nicht für den Datensatz relevante Zeichen wurden gelöscht.

Die Python Program war sehr einfach. Wir haben die Standartbibliothek 'csv' benutzt um ein csv_writer und csv_reader zu erzeugen. Wir haben dann die .csv Datei gelesen, und Zeile für Zeile haben wir geprüft ob der Tweet Truncated ist, oder abgeschnitten ist. Wenn nicht, haben wir den Tweet in einer anderen Datei *test.csv* gespeichert.

3.Aufgabe: Datenimport

4.Aufgabe: Webserver

Link zum Webserver:
https://github.com/gancia-kiss/dbs_projekt/tree/master/programs/server

Wir haben uns entschieden selbe einen Webserver zu entwickeln und zwar wollten wir die 'Flask' Framework von Python zu verwenden. Die Installation war sehr einfach, und es ist sehr einfach ein kurzes Programm zu entwickeln.

```
from flask import Flask

app = Flask(__name__)
port = 5234
host = '127.0.0.1'

@app.route('/')
def index():
    return 'Hello world'

if __name__ == '__main__':
    app.run(debug=True, host=host, port=port)
```

Das Programm erzeugt einen einfachen Webserver auf Port 5234. Wenn man die Seite *localhost:5234* öffnet, dann sieht man unformatiert 'Hello world'.

Es wäre sehr einfach das zu erweitern. Wir können .html Dateis im selben ordner legen, und dann neue Wege mit *@app.route()* definieren. Vielleicht werden wir die Programme für Hashtag-analyse in den Server einbauen, damit wir dynamisch den Inhalt anpassen können.