VILNIAUS UNIVERSITETAS

INFORMACINIŲ SISTEMŲ INŽINERIJA

PROJEKTINĖ UŽDUOTIS

DAŽNIAUSIAI TEKSTE PASITAIKANČIŲ ŽODŽIŲ PAIEŠKA

Darbą parengė: Gytautė Barzdžiūtė, Oskaras Diraitis, Dainius Masevičius, Tomas Klevas

2021, Vilnius

Turinys

Užduotis

Sukurkite programą, kuri rastų tekste dažniausiai pasitaikančius žodžius, išskyrus nereikšmingus žodžius. (angl. stop words)

Nuorodos

- Versijavimo sistema “Github” - https://github.com/gytautele/naturali_kalba
- Skaidrės - https://github.com/gytautele/naturali_kalba

Užduoties realizavimas

Tekstiniai failai

Šiai užduočiai atlikti naudojami 2 tekstiniai failai - “Stopwords.txt” ir “Text.txt”. “Stopwords” failo turinys yra toks: the, of, and, in, a, by, its, was, to, at, as, it, has, but, not. Tai nereikšmingi žodžiai, kurie nėra skaičiuojami kaip dažniausiai pasikartojantys žodžiai tekste. “Text” failo turinys - tai tekstas, kuris yra nagrinėjamas. Simboliška, kadangi pasirinktas tekstas apie Vilniaus universitetą anglų kalba. Dėl teksto ilgio šiame dokumente pateikiama tik jo ištrauka:

“History The University of Vilnius, one of the oldest and most famous establishments of higher education in Eastern and Central Europe, was founded in 1579. Functioning for a long time as the only school of higher learning in Lithuania, it was a preserver of cultural and scientific traditions, and has played a significant part in the cultural life not only of Lithuania, but the neighboring countries as well. <...> If you want to learn more about the history of Vilnius University, have a look at the publication Universitas Vilnensis 1579–2004 and Alma Mater Vilnensis.“

Bibliotekos

Užduočiai atlikti naudojamos 3 bibliotekos (žr. pav. nr. 1):

- import collections - importuojamos talpyklos, kuriose galima saugoti duomenis, jų kolekcijas (pavyzdžiui list, dict, set, tuple).
- import pandas - leidžia pateikti duomenis tokiu būdu, kuris yra tinkamas duomenų analizei

- import mathplotlib.pyplot - leidžia naudotis MATLAB programos privalumais: leidžia sukurti figūras, linijas, etiketes ir panašiai.

```
import collections
import pandas as pd
import matplotlib.pyplot as plt
```

Pav. nr. 1 Naudojamos bibliotekos

Programos kodas

Pirma, nuskaitomas failas, su nurodyta koduote (žr. pav. 2). UTF-8 koduotė - viena kintamo ilgio simbolių koduočių, kuria galima užrašyti bet kokį Unikodo simbolį.

```
# Read input file
# The encoding is specified here also
file = open('text.txt', encoding="utf8")
text = file.read()
```

Pav. nr. 2 Failo nuskaitymas, koduotė

Nustatomi žodžiai, kurie nebus skaičiuojami kaip dažniausiai pasikartojantys(žr. pav. 3). Tai vadinamieji “nereikšmingi žodžiai”. Vartotojas gali pats papildyti nereikšmingų žodžių sąrašą. Tai rodo programos lankstumą, tačiau atneša ir šiokių tokių trūkumų (pateikiama skyriuje “Testavimo išvados”).

```
# Stopwords
stopwords = set(line.strip() for line in open('stopwords.txt'))
stopwords = stopwords.union(set(['mr','mrs','one','two','said']))
```

Pav. nr. 3 Nereikšmingi žodžiai (angl. stop words)

Sukuriamas naudojamo teksto žodynas: jei žodis tame tekste yra naujas, jis įrašomas į žodyną, jeigu jau egzistuoja, padidinamas jo kiekis (žr. pav. 4, 5).

```
# Instantiate a dictionary, and for every word in the file,
# Add to the dictionary if it doesn't exist. If it does, increase the count.
wordcount = {}
```

Pav. nr. 4 Žodžių skaičiavimas

Norint eliminuoti dublikatus (pavyzdžiui Vilnius, Vilnius ir Vilnius:) pašalinami simboliai, sumažinamos didžiosios raidės. Sukuriamas ankstesniame žingsnyje minėtas teksto žodynas (žr. pav. 5).

```
# To eliminate duplicates, remember to split by punctuation, and use case demiliters.
for word in text.lower().split():
    word = word.replace(".","")
    word = word.replace(",","")
    word = word.replace(":","")
    word = word.replace("\"","")
    word = word.replace("!","")
    word = word.replace("â€œ","")
    word = word.replace("â€˜","")
    word = word.replace("*","")
    if word not in stopwords:
        if word not in wordcount:
            wordcount[word] = 1
        else:
            wordcount[word] += 1
```

Pav. nr. 5 Teksto "suvienodinimas", žodyno sukūrimas

Vartotojui leidžiama pasirinkti kiek dažniausiai pasikartojančių žodžių spausdinti. Atspausdinami rezultatai (žr. pav. 6, 7).

```
# Print most common word
n_print = int(input("How many most common words to print: "))
print("\nOK. The {} most common words are as follows\n".format(n_print))
word_counter = collections.Counter(wordcount)
for word, count in word_counter.most_common(n_print):
    print(word, ": ", count)
```

Pav. nr. 6 Rezultatų spausdinimo logika

```
How many most common words to print: 10

OK. The 10 most common words are as follows

university :  20
vilnius :  8
lithuania :  7
famous :  5
history :  4
soviet :  4
higher :  3
cultural :  3
vilnensis :  3
from :  3
```

Pav. nr. 7 Rezultatų spausdinimas

Pagal visas gerąsias programavimo praktikas uždarytą failą reikia uždaryti (žr. pav. 8).

```
# Close the file
file.close()
```

Pav. nr. 8 Failo uždarymas

Programinė įranga

Programos kūrimui, testavimui, rezultatų atvaizdavimui - projekto kūrimui pasirinkta naudoti Jupyter NoteBook. Tai patogus, greitas darbo įrankis, leidžiantis ne tik matyti tekstinius, tačiau ir vizualizuotus rezultatus.

Pav. nr. 9 Naudojama programinė įranga

Testavimo išvados

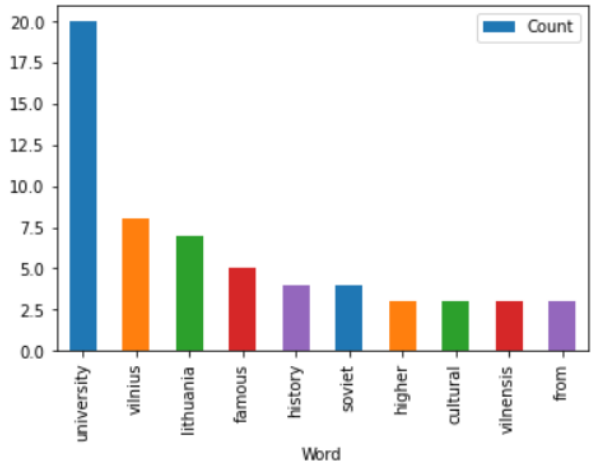
Programa gana nesudėtinga, tad gerai tvarkosi su nedidelės apimties tekstais, kurie naudoja pagrindinius skyrybos ženklus ir yra parašyti anglų kalba. Didelis programos trūkumas yra tas, kad kiekvieną skyrybos ženklą programoje reikia apibrėžti - didėja žmogiškos klaidos galimybė. Jeigu ši programa būtų naudojama didelės apimties tekstų analizei ir dažniausiai pasikartojančių žodžių paieškai, tikrai būtų ir klaidų. Dėl programos lankstumo galimybės sudaryti savo nereikšmingų žodžių sąrašą kyla ir pliusų ir minusų: vartotojas gali lanksčiai rinktis nereikšmingų žodžių rinkinį, jį sudaryti pagal save, tačiau dirbant su didelės apimties tekstais gali būti sunku apibrėžti visus norimus žodžius ir gali užtrukti daug laiko.

Rezultatų vizualizavimas

Programoje realizuota galimybė pasirinkti išvesties kiekį - kiek dažniausiai pasikartojančių žodžių atspausdinti. Pagal tai sudaroma diagrama, turinti 5 skirtingas spalvas (žr. pav. 10, 11). Jeigu atvaizduojami daugiau negu 5 stulpeliai, spalvos paeiliui kartojamos.

```
# Create a data frame of the most common words
# Draw a bar chart
lst = word_counter.most_common(n_print)
df = pd.DataFrame(lst, columns = ['Word', 'Count'])
df.plot.bar(x='Word',y='Count', color=['C0', 'C1', 'C2', 'C3', 'C4'])
```

Pav. nr. 10 Rezultatų vizualizavimas



Pav. nr. 11 Rezultatų vizualizavimas diagramoje