Sprawozdanie z laboratorium Teorii informacji i kompresji danych

Lab nr2 Przybliżanie języka naturalnego część II

Wojciech Szczepaniak

Index: 136808

Data zajęć: 18.12.2020 (czwartek 13:30)

1. Częstość słów

```
for i in data:
    if i not in word_ls:
        word_ls[i] = 1
    else:
        word_ls[i] +=1
word_ls = {k: v for k, v in sorted(word_ls.items(),key=lambda item: item[1],reverse=True)}
word_items = word_ls.items()
first_t = list(word_items)[:6000]
t_first = 0
for i in first_t:
    t_first += i[1]
```

Tym razem do obliczenia częstości wykorzystałem pythonowe słowniki ze względu na bardzo dużą ilość różnorodnych słów spowodowało to szybsze przetwarzanie.

Zbiór 30 tys. najpopularniejszych słów stanowił aż: 94,72%

A zbiór 6 tys. najpopularniejszych słów: 82,25%

2. Przybliżenie pierwszego rzędu

```
corpse = ""
word_items = list(word_ls)
for i in range(100):
    corpse = corpse +" "+ random.choices(word_items,word_ls.values())[0]
print(corpse)
```

Prawdopodobieństwo występowania słów została wyliczona na podstawie korpusu “norm_wiki_sample.txt”.

Wygenerowany ciąg 100 słów:

```
from vocals to better fadiora simone mall manager contractor uses the dropped the 11 primar
ily one lady coins men for regional the under hberlin their would international 3 gmina sta
nford actors apple content the way largest received threatening 1999 entered berlin l outdo
or women and d bridegroom decrease redesigned for scene plan a the various fell 29 in took
motivational metropolitans in atlanta behalf but beyond dorsey e the beast statistical club
s lower runs however the and supporting ernest american the ability jewish future however n
eeds some science after then city is the analog clayton 1637 450 south transferred morganti
```

3. Źródła Markova pierwszego rzędu na słowach

```
def p_creator(word,data):
    dick = {}
    counter = 0
    for i in range(len(data)-1):
        if data[i] == word:
            if data[i+1] not in dick:
                dick[data[i+1]] = 1

            else:
                dick[data[i+1]] += 1
            counter += 1
    for i in dick:
        dick[i] = dick[i] / counter
    return(dick)
```

```
for i in range(1):
    corpse = corpse +" "+ random.choices(word_items,word_ls.values())[0]

nested_dict = {}
new_word = corpse.replace(" ","")

for i in range(99):
    if new_word not in nested_dict:
        new_dict = p_creator(new_word, data)
        nested_dict[new_word] = new_dict
        new_word = random.choices(list(new_dict), new_dict.values())[0]
        corpse = corpse + " " + new_word
    else:
        old_dick = nested_dict[new_word]
        new_word = random.choices(list(old_dick), old_dick.values())[0]
        corpse = corpse + " " + new_word
```

Do wygenerowania prawdopodobieństw dla każdego słowa wykorzystano funkcję p_creator, która zliczała występujące słowa i zwracała słownik prawdopodobieństw. Następnie dany słownik był zapisywany w słowniku. Takie wykorzystanie zagnieżdżonych słowników powodowało przyspieszenie przetwarzania, gdyż algorytm nie musiał wielokrotnie liczyć tych samych prawdopodobieństw, jeżeli ostatnie słowo było już wcześniej wykorzystywane. Pierwsze słowo powstałego ciągu słów było wylosowane na podstawie prawdopodobieństw z poprzednich zadań.

Wygenerowany ciąg słów:

```
solo model sold in 2011 by the twenty two separate from the latvian and the hamlet in books
for three western australia entered budapest hungary the 1920s hardin and nearby star busma
nn 57 households out the republican era bloomed within five miles 6 280545 br d in 1949 pre
sident actress and 1974 and then twist he was lower field software is part of the median in
come received the town of roche pulled to be trained by gramotin had various duesenberg eng
ines included on 6 2014 tokyo auto tune breakdowns are only exists from northeastern corner
he worked with a german
```

4. Źródła Markova drugiego rzędu

```
def p_creator(word,data):
    dick = {}
    counter = 0
    word1,word2 = splitter(word)
    for i in range(len(data)-2):
        if data[i] == word1:
            if data[i+1] == word2:
                if data[i+2] not in dick:
                    dick[data[i+2]] = 1

                else:
                    dick[data[i+2]] += 1
                counter += 1
    for i in dick:
        dick[i] = dick[i] / counter
    return(dick)

def splitter(w_list):
    new_words = w_list.split()
    word1 = new_words[-2]
    word2 = new_words[-1]
    return(word1,word2)
```

```
for i in range(98):
    word1, word2 = splitter(corpse)
    new_word = merger(word1, word2)
    if new_word not in nested_dict:
        new_dict = p_creator(new_word, data)
        nested_dict[new_word] = new_dict
        if len(new_dict) == 0:
            new_dict = p_creator_uno(corpse, data)
            new_word = random.choices(list(new_dict), new_dict.values())[0]
            corpse = corpse + " " + new_word
        else:
            new_word = random.choices(list(new_dict), new_dict.values())[0]
            corpse = corpse + " " + new_word
    else:
        old_dick = nested_dict[new_word]
        new_word = random.choices(list(old_dick), old_dick.values())[0]
        corpse = corpse + " " + new_word
```

W tym wypadku wykorzystano funkcję bliźniaczo podobną do p_creator z różnicą, że dwu słowa zostały zapisane jako klucze słownika w postaci “słowo1 słowo2”. Oprócz tego należało zapewnić, że jeżeli po danym dwu słowie nie występowały żadne wyrazy to wygenerujemy losowe słowo na podstawie ostatniego słowa.

Początkowy string dwóch słów wygenerowano tworząc pierwsze słowo na podstawie prawdopodobieństwa z zad 1 a drugie na źródła Markova I rzędu.

Wygenerowany ciąg słów:

```
 temple yards s decision the producers were looking for their beauty and power steering wit
h optimized length and angle with high winds and bad boy reed forced to kill police officer
s have been included as the third time shih is never revealed his secret stockpile to benef
it from becoming members including the net effect of the formation of large scale british e
xpedition with 6 lines line a second guest star on the romanian music channels 10 minutes o
dd radio edit 4 00 a m commissioner rizzo and burk the accumulating liquid surrounding engi
ne 133 ignited immediately trapping the three continual councillors
```

Wygenerowany ciąg słów dla początkowego słowa “probability”:

```
probability bayes theorem likelihood functions linear models logistic models proportional h
azards models risk logistic models proportional hazards models risk logistic models models
proportional hazards models risk assessment that concluded that mankind can not serve direc
tly as candidate in the world egg at the end of the decade for the war ended it was propose
d as a child on awaji island he served on the uk eventually dropped alsip crestwood and bri
dgeview to its former french name service commun interprtation confrences is a local celebr
ity is mr ryan mcclure the school it consists of two railway stations it includes about 35
```