

2. Korzystając z zainstalowanego taggera otaguj morfosyntaktycznie i zlematyzuj wszystkie orzeczenia z wybranego roku.

```
url = 'http://localhost:9200'

def tag_judgment(data):
    global count
    for text in data['items']:
        if not reader.in_2008(text):
            continue
        response = requests.post(url, text['textContent'].encode('utf-8'))
        with open('tagged.txt', 'a') as f:
            f.writelines(response.text)
```

5. Oblicz statystykę logarytmiczną opartą o rozkład dwumienny dla tego zbioru.

```
def xLogX(x):
    if x == 0:
        return 0.0
    else:
        return x * math.log(x)

def entropy(*arg):
    sum = 0
    result = 0.0
    for element in arg:
        result += xLogX(element)
        sum += element

    return xLogX(sum) - result

def loglikelihoodRatio(k11, k12, k21, k22):
    rowEntropy = entropy(k11 + k12, k21 + k22)
    columnEntropy = entropy(k11 + k21, k12 + k22)
    matrixEntropy = entropy(k11, k12, k21, k22)
    if rowEntropy + columnEntropy < matrixEntropy:
        return 0.0
    return 2.0 * (rowEntropy + columnEntropy - matrixEntropy)
```

dla każdego bigramu wywołuję loglikelihoodRatio:

```
for bigram in count_bigrams.keys():
    w1, w2 = bigram.split(' ')
    k11 = count_bigrams[bigram]
    k12 = count_words[w2] - k11
    k21 = count_words[w1] - k11
    k22 = all_events - k11 - k12 - k21
    d[bigram] = loglikelihoodRatio(k11=k11, k12=k12, k21=k21, k22=k22)
```

count_bigrams to słownik gdzie kluczami są bigramy, a wartościami liczba ich wystąpień.

6. Przedstaw 30 pierwszych wyników, w których na pierwszy miejscu występuje rzeczownik, a na drugim miejscu rzeczownik lub przymiotnik.

('zamówienie:subst publiczny:adj', 178046.02098554373)
('sąd:subst wysoki:adj', 158704.91667073965)
('sąd:subst okręgowy:adj', 126726.58393001556)
('skarga:subst kasacyjny:adj', 124025.76081508398)
('trybunał:subst konstytucyjny:adj', 102516.57095706463)
('sąd:subst apelacyjny:adj', 93975.45056861639)
('ubezpieczenie:subst społeczny:adj', 71078.23536360264)
('dzień:subst styczeń:subst', 59304.822090923786)
('zero:subst grosz:subst', 57836.3045745492)
('urząd:subst zamówienie:subst', 55909.29310554266)
('skarb:subst państwo:subst', 51838.590127527714)
('sąd:subst rejonowy:adj', 47814.52246212959)
('dzień:subst lipiec:subst', 47422.18996155262)
('dzień:subst czerwiec:subst', 47191.763429522514)
('dzień:subst grudzień:subst', 45319.948646605015)
('kio:subst uzp:subst', 44467.19811797142)
('przedmiot:subst zamówienie:subst', 44380.25776797533)
('dzień:subst listopad:subst', 44038.22931319475)
('dzień:subst kwiecień:subst', 42444.29008048773)
('wyrok:subst sąd:subst', 42150.14617139101)
('dzień:subst październik:subst', 41404.12111568451)
('działalność:subst gospodarczy:adj', 41391.84367316961)
('posiedzenie:subst niejawny:adj', 39143.581896066666)
('prawo:subst zamówienie:subst', 38990.345791220665)
('dzień:subst maj:subst', 38234.97927439213)
('rachunek:subst dochód:subst', 38149.510890603065)
('dzień:subst marzec:subst', 37295.34700548649)
('dzień:subst wrzesień:subst', 36987.536076545715)
('dochód:subst własny:adj', 36493.86270236969)
('izba:subst odwoławczy:adj', 36115.94044327736)