Lab4

Zad1

In [252…
```
from unidecode import unidecode
import spacy
from spacy.tokenizer import Tokenizer
import random
```

In [101…
```
def sigma(a,b):
    return not a == b

def edit_length(x, y, f_sigma):
    m, n = len(x), len(y)
    edit = [[0]*(n+1) for i in range(m+1)]

    edit[0][0] = (0, "Nothing")
    for i in range(1,m+1):
        edit[i][0] = (i, 1)
    for j in range(1,n+1):
        edit[0][j] = (j, 2)

    if n == 0 and m == 0:
        return edit[0][0][0], edit[0][0][1]

    # 1 delete , 2 insert, 3 change, 4 pass
    for i in range(1,m+1):
        for j in range(1,n+1):
            a = edit[i-1][j][0] + 1
            b = edit[i][j-1][0] + 1
            change_value = f_sigma(x[i-1], y[j-1])
            c = edit[i-1][j-1][0] + change_value
            if a < b and a < c:
                edit[i][j] = (a, 1)
            elif b < c:
                edit[i][j] = (b, 2)
            else:
                if change_value:
                    edit[i][j] = (c, 3)
                else:
                    edit[i][j] = (c, 4)

    path = []
    i = m
    j = n
    while i != 0 or j != 0:
        path_value = edit[i][j][1]
        if path_value == 1:
            path.append("Delete")
            i -= 1
        elif path_value == 2:
            path.append("Insert")
            j -= 1
        elif path_value == 3:
            path.append("Change")
            i -= 1
            j -= 1
        else:
            path.append("Pass")
            i -= 1
            j -= 1

    path.reverse()
    return edit[m][n][0], path
```

Zad2

In [7]:
```
def x2y_visualisation(x, y):
    length, path = edit_length(x, y, sigma)
    print(path)
    x_idx = 0
    y_idx = 0
    print("{} => {}".format(x, y))
    for action in path:
        if action == "Insert":
            print("{}*{}*{} insert {}".format(x[0:x_idx],y[y_idx], x[x_idx:], y[y_idx]))
            x = x[0:x_idx] + y[y_idx] + x[x_idx:]
        elif action == "Delete":
            print("{}*{}*{} delete {}".format(x[0:x_idx], x[x_idx], x[x_idx+1:], x[x_idx]))
            x = x[0:x_idx] + x[x_idx+1:]
```

```
            x_idx -= 1
            y_idx -= 1
        elif action == "Change":
            print("{}*{}*{} Change {} -> {}".format(x[0:x_idx], y[y_idx], x[x_idx+1:], x[x_idx], y[y_idx]))
            x = x[0:x_idx] + y[y_idx] + x[x_idx+1:]
        x_idx += 1
        y_idx += 1
    print(x)
```

Zad3

In [161…
```
x2y_visualisation("los", "kloc")
```

```
['Insert', 'Pass', 'Pass', 'Change']
los => kloc
*k*los insert k
klo*c* Change s -> c
kloc
```

In [9]:
```
x2y_visualisation("Łódź", "Lodz")
```

```
['Change', 'Change', 'Pass', 'Change']
Łódź => Lodz
*L*ódź Change Ł -> L
L*o*dź Change ó -> o
Lod*z* Change ź -> z
Lodz
```

In [10]:
```
x2y_visualisation("kwintesencja", "quintessence")
```

```
['Change', 'Change', 'Pass', 'Pass', 'Pass', 'Pass', 'Insert', 'Pass', 'Pass', 'Pass', 'Pass', 'Delete', 'Change'
]
kwintesencja => quintessence
*q*wintesencja Change k -> q
q*u*intesencja Change w -> u
quinte*s*sencja insert s
quintessenc*j*a delete j
quintessenc*e* Change a -> e
quintessence
```

In [11]:
```
x2y_visualisation("ATGAATCTTACCGCCTCG", "ATGAGGCTCTGGCCCCTG")
```

```
['Pass', 'Pass', 'Pass', 'Pass', 'Change', 'Change', 'Pass', 'Pass', 'Insert', 'Pass', 'Change', 'Change', 'Pass'
, 'Change', 'Pass', 'Pass', 'Pass', 'Delete', 'Pass']
ATGAATCTTACCGCCTCG => ATGAGGCTCTGGCCCCTG
ATGA*G*TCTTACCGCCTCG Change A -> G
ATGAG*G*CTTACCGCCTCG Change T -> G
ATGAGGCT*C*TACCGCCTCG insert C
ATGAGGCTCT*G*CCGCCTCG Change A -> G
ATGAGGCTCTG*G*CGCCTCG Change C -> G
ATGAGGCTCTGGC*C*CCTCG Change G -> C
ATGAGGCTCTGGCCCCT*C*G delete C
ATGAGGCTCTGGCCCCTG
```

Zad4

Computing subsequence, x_diff, y_diff

In [151…
```
def lcs(x, y):
    m, n = len(x), len(y)
    edit = [[0]*(n+1) for i in range(m+1)]

    edit[0][0] = (0, "")
    for i in range(1,m+1):
        edit[i][0] = (0, 1)
    for j in range(1,n+1):
        edit[0][j] = (0, 2)

    if n == 0 and m == 0:
        return edit[0][0][0], edit[0][0][1]

    # 1 delete , 2 insert, 3 change,4 append lcs (diagonal)
    for i in range(1,m+1):
```

```
        for j in range(1,n+1):
            a = edit[i-1][j][0]
            b = edit[i][j-1][0]
            change_value = (x[i-1] == y[j-1])
            c = edit[i-1][j-1][0] + change_value
            if a > b and a > c:
                edit[i][j] = (a, 1)
            elif b > c:
                edit[i][j] = (b, 2)
            else:
                if change_value:
                    edit[i][j] = (c, 4)
                else:
                    edit[i][j] = (c, 3)
    lcs = []
    x_diff = []
    y_diff = []
    i = m
    j = n
    while i != 0 or j != 0:
        path_value = edit[i][j][1]
        if path_value == 1:
            x_diff.append(x[i-1])
            i -= 1
        elif path_value == 2:
            y_diff.append(y[j-1])
            j -= 1
        elif path_value == 3:
            x_diff.append(x[i-1])
            y_diff.append(y[j-1])
            i -= 1
            j -= 1
        elif path_value == 4:
            lcs.append(y[j-1])
            i -= 1
            j -= 1

    lcs.reverse()
    x_diff.reverse()
    y_diff.reverse()
    return edit[m][n][0], "".join(lcs), "".join(x_diff), "".join(y_diff)
```

In [354…
```
lcs("rower", "oxwe")
```

Out[354… `(3, 'owe', 'rr', 'x')`

Computing length

In [14]:
```
def lcs_sigma(a, b):
    if a == b:
        return 0
    else:
        return 2

def lcs_length(x, y):
    return int((len(x) + len(y) - edit_length(x,y,lcs_sigma)[0]) / 2)
```

In [213…
```
lcs_length("rower", "oxwe")
```

Out[213… `3`

Zad5

Read file

In [352…
```
with open("romeo-i-julia-700.txt", "r", encoding="utf8") as file:
    lines = file.readlines() # lines

nlp = spacy.load("pl_core_news_sm")

# Tokenize text line by line
tokens_1 = [[] for i in range(len(lines))]
tokens_2 = [[] for i in range(len(lines))]
tokens_number = 0
n = len(lines)
for i in range(n):
    line_tokens = list(nlp.tokenizer(lines[i]))
```

```
    tokens_1[i] = line_tokens.copy()
    tokens_2[i] = line_tokens.copy()
    tokens_number += len(line_tokens)
```

In [353…
```
print("Number of tokens : ", tokens_number)
```

```
Number of tokens :  3053
```

Remove 3% of tokens

In [349…
```
tokens_number_after_remove = int(0.97 * tokens_number)
while tokens_number != tokens_number_after_remove:
    i = random.randrange(len(lines))
    j = random.randrange(len(lines))

    n = len(tokens_1[i])
    m = len(tokens_2[j])

    if n == 0 or m == 0:
        continue

    k = random.randrange(n)
    l = random.randrange(m)

    if tokens_1[i][k].text == '\n' or tokens_2[j][l].text == '\n':
        continue

    tokens_1[i].pop(k)
    tokens_2[j].pop(l)

    tokens_number -= 1
```

Longest common subsequence of tokens

In [338…
```
t_1 = []
t_2 = []
for i in range(len(lines)):
    t_1 += tokens_1[i]
    t_2 += tokens_2[i]

lcs_length(t_1, t_2)
```

Out[338… 2877

Zad8

In [340…
```
def print_diff_line(diff, index, file):
    if len(diff) > 0:
        if diff[len(diff) - 1] == '\n': # remove end of line in diff if n < m
            diff = diff[:-1]
        if len(diff) > 0:
            print("<line:{}, '{}'>:{}".format(index, file, diff))

def diff(file_1, file_2):
    with open(file_1, "r", encoding="utf8") as f:
        lines_1 = f.readlines()

    with open(file_2, "r", encoding="utf8") as f:
        lines_2 = f.readlines()

    n = len(lines_1)
    m = len(lines_2)


    for i in range(n):
        if i < m:
            lcs_length, lcs_subsequence, x_diff, y_diff = lcs(lines_1[i], lines_2[i])
            print_diff_line(x_diff, i, file_1)
            print_diff_line(y_diff, i, file_2)
        else:
            print_diff_line(lines_1[i], i, file_1) # print rest of the first file

    for i in range(n, m):
        print_diff_line(lines_2[i], i, file_2) # print rest of the second file
```

text_1.txt:

rower

abcdefgh

jojojo

text_2.txt:
oxwe
bcxxdefgh
ojyyjo
ergAGE

In [341…
```
diff("text_1.txt", "text_2.txt")
```

```
<line:0, 'text_1.txt'>:rr
<line:0, 'text_2.txt'>:x
<line:1, 'text_1.txt'>:a
<line:1, 'text_2.txt'>:xx
<line:2, 'text_1.txt'>:jo
<line:2, 'text_2.txt'>:yy
<line:3, 'text_2.txt'>:ergAGE
```

Zad9

Write tokenized text to files

In [350…
```
with open("tokens_1.txt", "w+", encoding="utf8") as f:
    for line_tokens in tokens_1:
        f.write(" ".join(map(str,line_tokens)))

with open("tokens_2.txt", "w+", encoding="utf8") as f:
    for line_tokens in tokens_2:
        f.write(" ".join(map(str,line_tokens)))
```

In [351…
```
diff("tokens_1.txt", "tokens_2.txt")
```

```
<line:2, 'tokens_2.txt'>:Romeo
<line:5, 'tokens_1.txt'>:88 -
<line:5, 'tokens_2.txt'>: -903
<line:11, 'tokens_2.txt'>: szlachetnego
<line:13, 'tokens_1.txt'>:
<line:13, 'tokens_2.txt'>: —
<line:17, 'tokens_2.txt'>: —
<line:18, 'tokens_1.txt'>: *
<line:20, 'tokens_1.txt'>:
<line:21, 'tokens_1.txt'>:
<line:25, 'tokens_1.txt'>:
<line:28, 'tokens_1.txt'>: Montekiego
<line:29, 'tokens_2.txt'>: KAPULET
<line:51, 'tokens_1.txt'>: z
<line:53, 'tokens_2.txt'>: rodzicielskie
<line:57, 'tokens_2.txt'>: ,
<line:58, 'tokens_1.txt'>: przedstawienia
<line:61, 'tokens_2.txt'>: …
<line:82, 'tokens_1.txt'>: bobyśmy
<line:97, 'tokens_2.txt'>: zwyczaj
<line:117, 'tokens_1.txt'>:a każdego
<line:127, 'tokens_2.txt'>: od
<line:132, 'tokens_2.txt'>:t ylko
<line:142, 'tokens_1.txt'>:z predsiębrać
<line:157, 'tokens_1.txt'>:SAMSON
<line:159, 'tokens_1.txt'>: dobyty .
<line:162, 'tokens_1.txt'>:GRZEGORZ
<line:179, 'tokens_1.txt'>: za
<line:184, 'tokens_2.txt'>: ;
<line:187, 'tokens_2.txt'>:SAMSON
<line:194, 'tokens_2.txt'>:i pane
<line:204, 'tokens_2.txt'>:Czy
<line:211, 'tokens_2.txt'>: jest
<line:216, 'tokens_2.txt'>: .
<line:221, 'tokens_1.txt'>:ie skrzywłm
<line:226, 'tokens_2.txt'>: Abrahama
<line:233, 'tokens_1.txt'>: nie .
<line:248, 'tokens_1.txt'>: będzie
<line:260, 'tokens_1.txt'>:SAMSON
<line:265, 'tokens_2.txt'>:ABRAHAM
<line:270, 'tokens_1.txt'>:SAMSON
<line:272, 'tokens_2.txt'>: oswim
<line:279, 'tokens_2.txt'>: /
<line:281, 'tokens_1.txt'>: /
<line:286, 'tokens_2.txt'>:Cóż
```

<line:293, 'tokens_2.txt'>: nim
<line:301, 'tokens_1.txt'>: nikczemny
<line:303, 'tokens_2.txt'>: .
<line:309, 'tokens_1.txt'>:z
<line:311, 'tokens_2.txt'>: iPan
<line:325, 'tokens_2.txt'>:KAPULET
<line:327, 'tokens_1.txt'>: !
<line:327, 'tokens_2.txt'>: Monteki
<line:328, 'tokens_1.txt'>: .
<line:328, 'tokens_2.txt'>: szydnie
<line:335, 'tokens_1.txt'>: !
<line:352, 'tokens_1.txt'>: stali
<line:352, 'tokens_2.txt'>:Bezcześciciele
<line:353, 'tokens_1.txt'>:Czy
<line:358, 'tokens_2.txt'>:e tgo
<line:363, 'tokens_2.txt'>: poważni
<line:366, 'tokens_2.txt'>:e dłoni
<line:367, 'tokens_2.txt'>:y zardzewiałm
<line:369, 'tokens_1.txt'>:ś wań
<line:369, 'tokens_2.txt'>: ,
<line:370, 'tokens_2.txt'>:cie żym
<line:373, 'tokens_1.txt'>: zaś
<line:375, 'tokens_1.txt'>:a m
<line:377, 'tokens_1.txt'>: śmierci
<line:379, 'tokens_1.txt'>: obywatele
<line:385, 'tokens_1.txt'>: ?
<line:390, 'tokens_2.txt'>:naszego
<line:394, 'tokens_1.txt'>: zionąc
<line:395, 'tokens_2.txt'>: nim
<line:396, 'tokens_2.txt'>:Które
<line:397, 'tokens_1.txt'>: zamachów
<line:400, 'tokens_2.txt'>: i
<line:403, 'tokens_2.txt'>:NI MOTEK
<line:405, 'tokens_1.txt'>: Romeo
<line:406, 'tokens_2.txt'>: ,
<line:412, 'tokens_1.txt'>:się
<line:412, 'tokens_2.txt'>:oknach
<line:414, 'tokens_2.txt'>: ów
<line:417, 'tokens_2.txt'>:Ledwiem
<line:419, 'tokens_1.txt'>: w
<line:421, 'tokens_1.txt'>: (
<line:422, 'tokens_2.txt'>:)
<line:424, 'tokens_2.txt'>: w
<line:425, 'tokens_1.txt'>:me
<line:425, 'tokens_2.txt'>:ukał
<line:436, 'tokens_1.txt'>: ,
<line:437, 'tokens_1.txt'>: pokoju
<line:438, 'tokens_2.txt'>: dnia
<line:446, 'tokens_2.txt'>: powód
<line:451, 'tokens_2.txt'>: niego
<line:463, 'tokens_1.txt'>: w
<line:466, 'tokens_1.txt'>: swój kielich
<line:466, 'tokens_2.txt'>:Nim
<line:467, 'tokens_1.txt'>: przed
<line:469, 'tokens_1.txt'>:o śrdka
<line:471, 'tokens_1.txt'>: głębi
<line:476, 'tokens_1.txt'>:o nadchdzi
<line:480, 'tokens_2.txt'>:MONTEKI
<line:488, 'tokens_2.txt'>:BENWOLIO
<line:490, 'tokens_2.txt'>: .
<line:495, 'tokens_1.txt'>:   nie
<line:500, 'tokens_2.txt'>:a bił
<line:503, 'tokens_1.txt'>:ROMEO
<line:507, 'tokens_1.txt'>: zboczyli
<line:510, 'tokens_2.txt'>:BENWOLIO
<line:512, 'tokens_1.txt'>: dłuży
<line:525, 'tokens_1.txt'>:ROMEO
<line:532, 'tokens_2.txt'>: to
<line:535, 'tokens_1.txt'>:ROMEO
<line:537, 'tokens_1.txt'>: jej
<line:537, 'tokens_2.txt'>: skąd
<line:549, 'tokens_1.txt'>:e cl
<line:550, 'tokens_1.txt'>:jeć
<line:550, 'tokens_2.txt'>:dzi
<line:552, 'tokens_1.txt'>: imłość
<line:553, 'tokens_2.txt'>: !
<line:557, 'tokens_1.txt'>: chaosie
<line:558, 'tokens_2.txt'>: !
<line:560, 'tokens_2.txt'>:Taką
<line:561, 'tokens_1.txt'>: się
<line:566, 'tokens_2.txt'>: ,
<line:576, 'tokens_1.txt'>: ,
<line:577, 'tokens_2.txt'>: duszy

<line:583, 'tokens_2.txt'>: przez
<line:585, 'tokens_2.txt'>:ię powkszasz
<line:588, 'tokens_2.txt'>: Miłość
<line:591, 'tokens_1.txt'>: którym tonie
<line:593, 'tokens_2.txt'>:Żółcią
<line:596, 'tokens_1.txt'>:e odjść
<line:596, 'tokens_2.txt'>:/
<line:608, 'tokens_1.txt'>: co
<line:613, 'tokens_1.txt'>: to
<line:616, 'tokens_2.txt'>:ROMEO
<line:629, 'tokens_2.txt'>:ROMEO
<line:637, 'tokens_2.txt'>:BENWOLIO
<line:639, 'tokens_1.txt'>:   m ierzył
<line:654, 'tokens_2.txt'>:ROMEO
<line:657, 'tokens_1.txt'>: ;
<line:658, 'tokens_1.txt'>:d twarą
<line:658, 'tokens_2.txt'>: wstydliwości swojej
<line:659, 'tokens_2.txt'>: się
<line:660, 'tokens_1.txt'>: ;
<line:660, 'tokens_2.txt'>:Szydzi
<line:662, 'tokens_1.txt'>:e zjdna .
<line:663, 'tokens_2.txt'>:ie bdna
<line:668, 'tokens_1.txt'>:BENWOLIO
<line:677, 'tokens_2.txt'>:Całą
<line:679, 'tokens_1.txt'>: mądrze
<line:680, 'tokens_2.txt'>: nie
<line:686, 'tokens_1.txt'>: na
<line:691, 'tokens_1.txt'>:Doradź
<line:697, 'tokens_2.txt'>:c ozom
<line:698, 'tokens_2.txt'>:Rozpatrywania