# МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА**

# кафедра інформаційних систем та технологій

**ЗВІТ**

із лабораторної роботи № 1

з дисципліни **«Бази даних та бази знань»**

на тему: «Автоматичне укладання частотного словника»

Виконав: Студент групи 2 Шидей Роман

Перевірив:

# Київ — 2023

Хід роботи

ВСТУП

1.1 Опис 2-ох вибірок за параметрами лінгвістичної однорідності

# Вибірка №1

Текст: «Виклики національного самовизначення»

Автор: Розумний М. М.

Рік видання: 2016

Стиль: науковий текст

Жанр: монографія

# Вибірка №2

Назва тексту: «Амулет Паскаля»

Автор: Ірен Роздобудько

Рік видання: 2009

Стиль: художній

Жанр: роман

1.2.Опис 2-ох вибірок за параметрами статистичної однорідності:

|  |  |  |
| --- | --- | --- |
|  | Вибірка 1 | Вибірка 2 |
| Метод організації вибірки | механічний | механічний |
| Обсяг вибірки | Перші 10000 словоформ із 48000+  Ця вибірка вже є частково опрацьованою (для кращої токенізації забрана передмова та зміст, посилання на російськомовні джерела, інформація на форзацах, тощо) | Перші 10000 словоформ із 35000+ |

ОСНОВНА ЧАСТИНА (код з коментарями)

import sqlite3

from sklearn.feature\_extraction.text import TfidfVectorizer

import pymorphy3

from collections import Counter, OrderedDict

import re

conn = sqlite3.connect('lab.db')

c = conn.cursor()

# Означення функції для створення таблиць із різних текстів

def create\_tables(text\_number):

c.execute(f"""CREATE TABLE IF NOT EXISTS Main\_text{text\_number}

(word\_id INTEGER PRIMARY KEY AUTOINCREMENT,

word\_form TEXT,

lemma TEXT,

part\_of\_speech TEXT,

word\_form\_frequency INTEGER)""")

c.execute(f"""CREATE TABLE IF NOT EXISTS WordFormsFreq\_text{text\_number}

(word\_id INTEGER PRIMARY KEY AUTOINCREMENT,

word\_form TEXT,

word\_form\_frequency INTEGER)""")

c.execute(f"""CREATE TABLE IF NOT EXISTS LemmasFreq\_text{text\_number}

(lemma\_id INTEGER PRIMARY KEY AUTOINCREMENT,

lemma TEXT,

lemma\_frequency INTEGER)""")

c.execute(f"""CREATE TABLE IF NOT EXISTS PosFreq\_text{text\_number}

(pos\_id INTEGER PRIMARY KEY AUTOINCREMENT,

part\_of\_speech TEXT,

pos\_frequency INTEGER)""")

# Означення функції для вибору файлу та його токенізації

def text\_choice(file\_path):

with open(rf"{file\_path}", 'r', encoding = "utf-8") as file:

global word\_dict, word\_list

word\_dict = {}

word\_list = []

pattern = r"[ҐґЄєІіЇїА-Яа-я’'-]+"

text\_1 = file.read().lower()

splitted\_1 = re.findall(pattern, text\_1)

number\_of\_tokens = 10000

for token in splitted\_1:

if number\_of\_tokens > 0:

if token not in word\_dict.keys():

word\_dict.update({token: 1})

word\_list.append(token)

number\_of\_tokens -= 1

else:

word\_list.append(token)

word\_dict[token] += 1

number\_of\_tokens -= 1

else: break

morph = pymorphy3.MorphAnalyzer(lang='uk')

# Означення функції для отримання та збереження даних із тексту, як список лем, частоту словоформ та лем, відповідність частині мови і т.д.

def create\_info(dict):

global lemmas

word\_forms, frequencies, word\_ids, lemmas, pos = [], [], [], [], []

id\_number = 1

for key, value in dict.items():

word\_ids.append(id\_number)

id\_number += 1

word\_forms.append(key)

frequencies.append(value)

word = morph.parse(key)[0]

lemmas.append(word.normal\_form)

pos.append(word.tag.POS)

main\_info = [tuple(word\_ids), tuple(word\_forms),

tuple(lemmas), tuple(pos), tuple(frequencies)]

word\_forms\_info = [tuple(word\_ids), tuple(word\_forms), tuple(frequencies)]

lemmas, pos = [], []

for token in word\_list:

word = morph.parse(token)[0]

lemmas.append(word.normal\_form)

pos.append(word.tag.POS)

lemma\_counts = Counter(lemmas)

lemma\_info = [

tuple(range(1, len(lemma\_counts) + 1)),

tuple(lemma\_counts.keys()),

tuple(lemma\_counts.values())]

pos\_counts = Counter(pos)

pos\_info = [

tuple(range(1, len(pos\_counts) + 1)),

tuple(pos\_counts.keys()),

tuple(pos\_counts.values())]

return main\_info, word\_forms\_info, lemma\_info, pos\_info

# Означення функції для збереження інформації в базі даних

def db\_update(text\_number, m\_info, wf\_info, l\_info, p\_info):

transposed\_info = list(zip(\*m\_info))

c.executemany(f"""INSERT INTO Main\_text{text\_number} VALUES

(?, ?, ?, ?, ?)""", transposed\_info)

transposed\_info = list(zip(\*wf\_info))

c.executemany(f"""INSERT INTO WordFormsFreq\_text{text\_number} VALUES

(?, ?, ?)""", transposed\_info)

transposed\_info = list(zip(\*l\_info))

c.executemany(f"""INSERT INTO LemmasFreq\_text{text\_number} VALUES

(?, ?, ?)""", transposed\_info)

transposed\_info = list(zip(\*p\_info))

c.executemany(f"""INSERT INTO PosFreq\_text{text\_number} VALUES

(?, ?, ?)""", transposed\_info)

conn.commit()

# Означення функції для отримання TF-IDF та його збереження в базі даних

def tf\_idf(text\_number, lemmas):

tfidf\_vectorizer = TfidfVectorizer()

tfidf\_matrix = tfidf\_vectorizer.fit\_transform([' '.join(lemmas)])

features\_names = tfidf\_vectorizer.get\_feature\_names\_out()

tfidf\_scores = tfidf\_matrix.toarray()[0]

c.execute(f"""CREATE TABLE IF NOT EXISTS TF\_IDF\_text{text\_number}

(lemma\_id INTEGER PRIMARY KEY AUTOINCREMENT,

lemma TEXT,

tf\_idf REAL)""")

tf\_idf\_list = []

for i, a in enumerate(features\_names):

tf\_idf\_list.append((features\_names[i], tfidf\_scores[i] ))

c.executemany(f"""INSERT INTO TF\_IDF\_text{text\_number}

(lemma, tf\_idf) VALUES (?, ?)""", tf\_idf\_list)

conn.commit()

# Запуск функцій

create\_tables(1)

text\_choice(r"D:\Roma\Student\2023-2024\Традиційна та комп'ютерна лексикографія\rozumniy-druk-915a0-redacted.txt")

main\_info, word\_forms\_info, lemma\_info, pos\_info = create\_info(word\_dict)

db\_update(1, main\_info, word\_forms\_info, lemma\_info, pos\_info)

tf\_idf(1, lemmas)

create\_tables(2)

text\_choice(r"D:\Roma\Student\2023-2024\Традиційна та комп'ютерна лексикографія\amulet-paskalia – копія.txt")

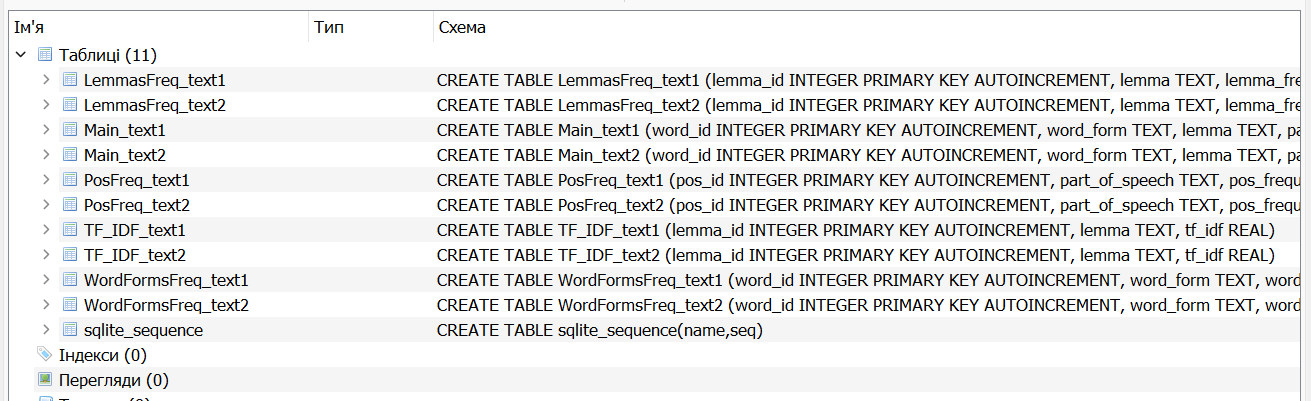
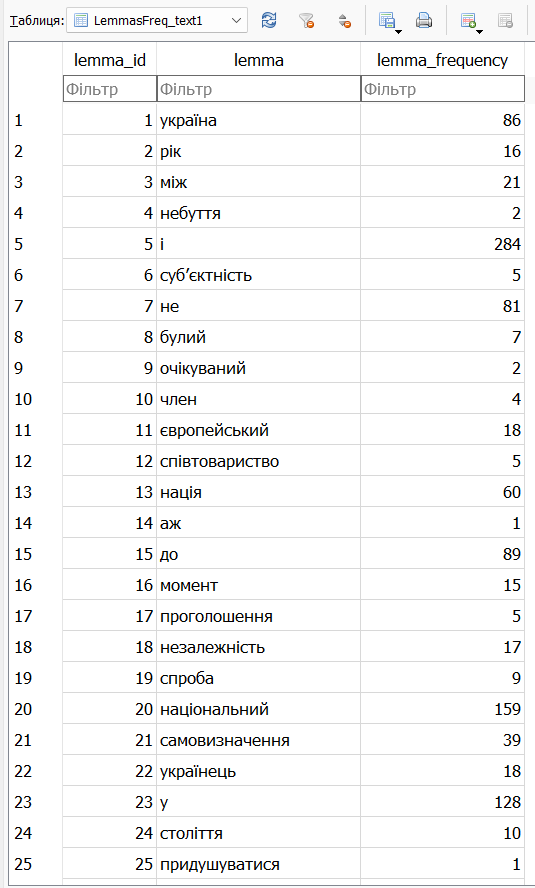
main\_info, word\_forms\_info, lemma\_info, pos\_info = create\_info(word\_dict)

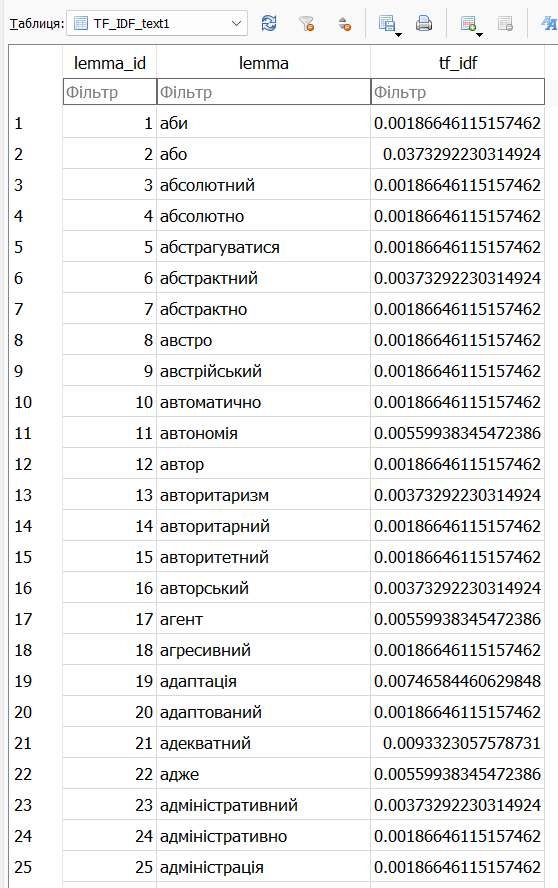
db\_update(2, main\_info, word\_forms\_info, lemma\_info, pos\_info)

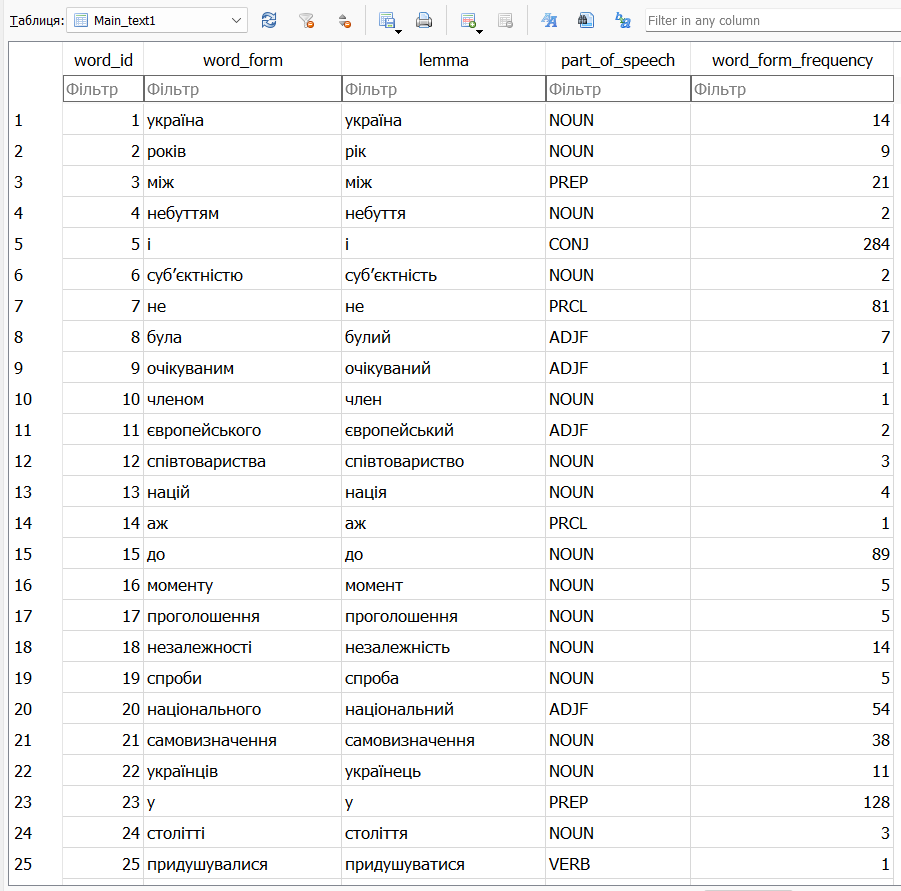
tf\_idf(2, lemmas)

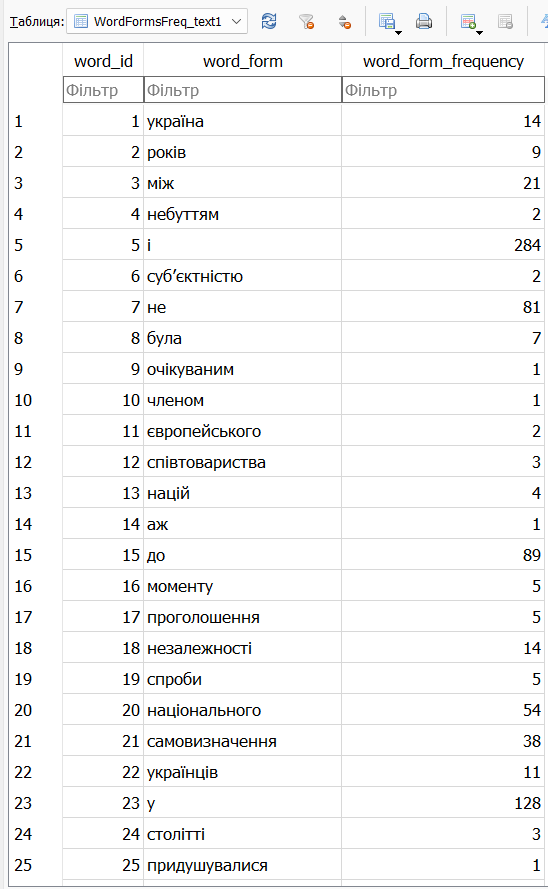
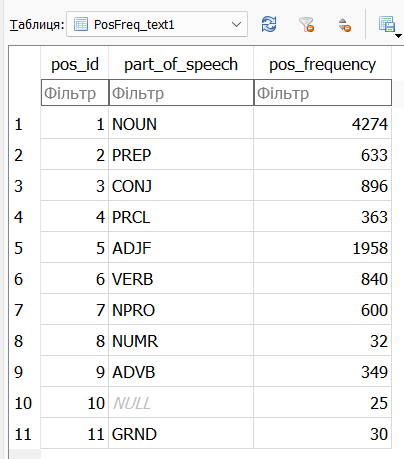
conn.close()

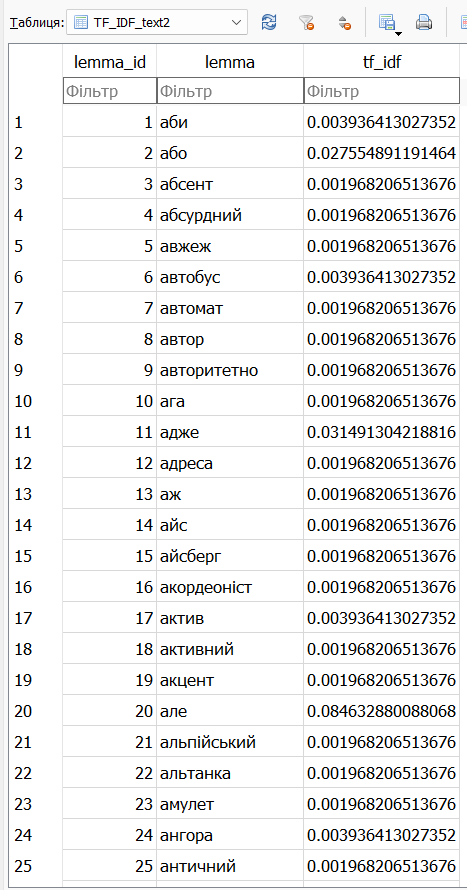
Список таблиць, створених кодом:

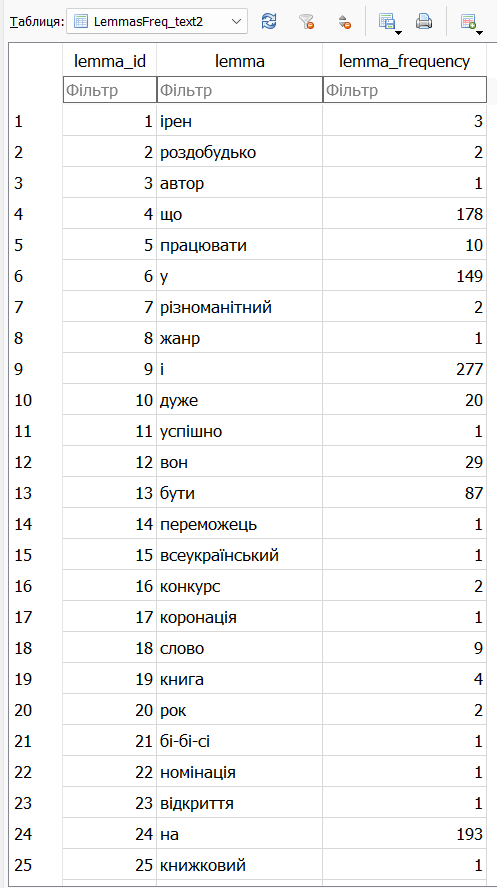
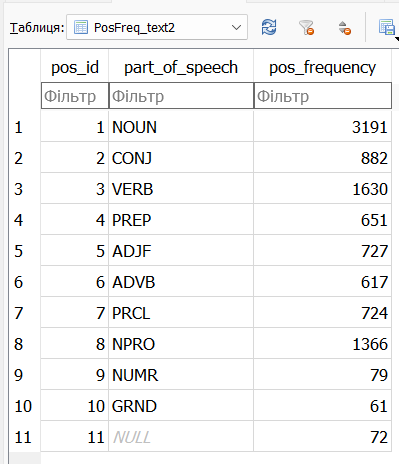
Результат роботи коду для вибірки №1 у БД:

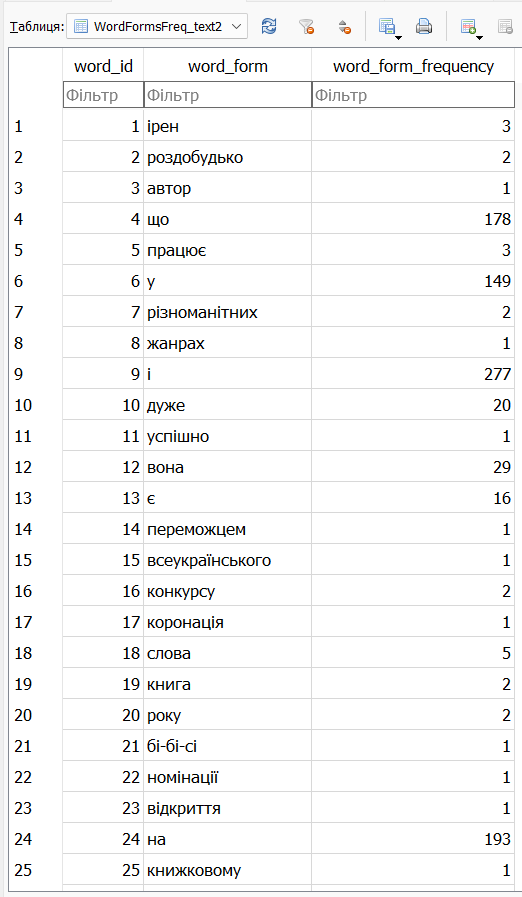
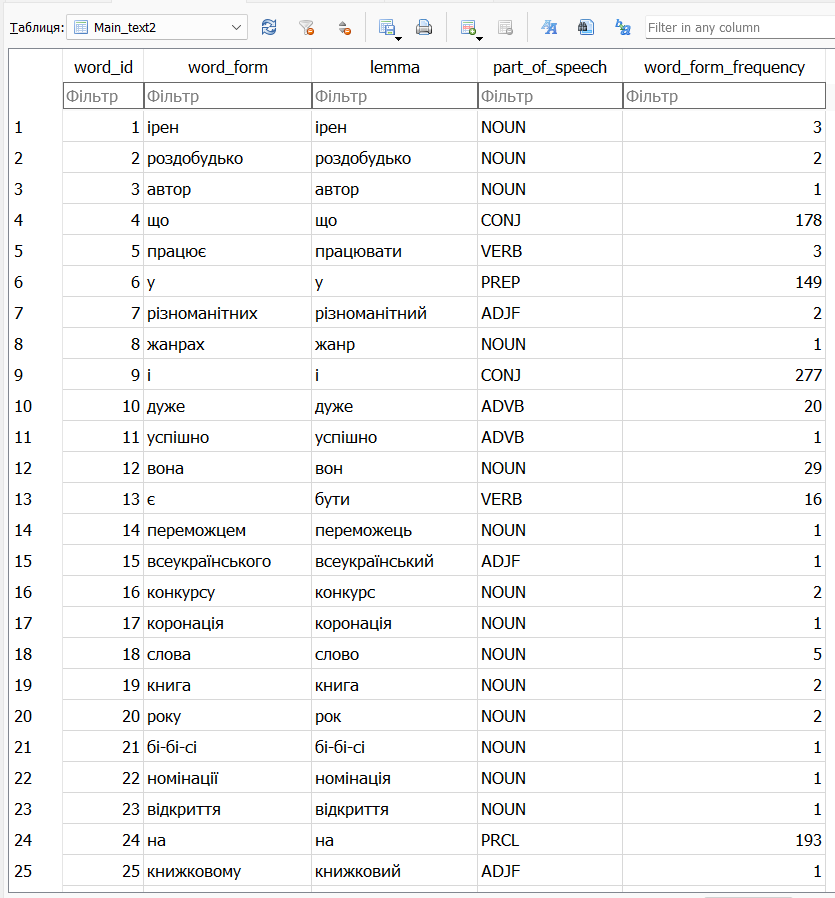






Результати роботи коду для 2 вибірки:





# Висновки

У результаті виконання лабораторної роботи було створено базу даних із шістьма частотними словниками та двома проміжними таблицями на основі двох текстових вибірок, де кожна вибірка має по 3 словники та проміжній таблиці із 10 тисячами слововживань. Для токенізації використані цикли та регулярні вирази, для лематизації – бібліотека pymorphy3 для української мови. Було обчислено та додано до бази даних статистику TF-IDF для кожної текстової вибірки.