## МИНОБРНАУКИ РОССИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»**

Кафедра теоретических основ компьютерной безопасности и криптографии

Теория информации

ЛАБОРАТОРНАЯ РАБОТА №1

студентки 3 курса 331 группы  
направления 10.05.01 – Компьютерная безопасность

факультета компьютерных наук и информационных технологий

Чуйкиной Алёны Сергеевны

Преподаватель – А.А.Лобов 16.10.2025

Саратов – 2025 г.

## Задание 1. Сбор корпуса текстов (общее задание)

**Постановка задачи:**

Найдите большой источник текстов. Это может быть Википедия, это могут быть книги. Это могут быть статьи с блогов и сайты.

Загрузите их и сформируйте большой корпус текстов. Должно быть не менее 64 мегабайт текста.

Отчётность: файл с описанием текстов, которые были лично получены. Указать скриншот, в котором показан размер данных.

Будем использовать в качестве источника текста страницы Википедии о российских городах.

**Алгоритм:**

1. Получить список ссылок на городские статьи со страницы "Список городов России".

2. Для каждой ссылки скачать HTML и извлечь видимый текст.

3. Сохранить каждый текст в файл в папке cities/.

4. Объединить файлы в corpus\_cities\_raw.txt.

5. Посчитать байты, слова, уникальные слова, энтропию, gzip-размер.

6. Сделать скриншот с подтверждением объёма (>=64 МБ).

**Листинг программы с описанием системы программирования.**

* Язык: Python 3.12
* Среда выполнения: Windows 11
* Среда разработки: IDLE
* Установленные библиотеки: requests — для загрузки страниц, beautifulsoup4 — для извлечения текста из HTML

Ссылка на git hub: https://github.com/seachna/information\_theory

get\_city\_links.py:

import requests

from bs4 import BeautifulSoup

from urllib.parse import urljoin

BASE = "https://ru.wikipedia.org"

LIST\_URL = "https://ru.wikipedia.org/wiki/Список\_городов\_России"

resp = requests.get(LIST\_URL, headers={"User-Agent":"CorpusBot/1.0 (student@example.com)"})

resp.raise\_for\_status()

soup = BeautifulSoup(resp.text, "html.parser")

links = set()

# На странице есть таблицы с ссылками — находим все <a> внутри статьи, фильтруем внутренние ссылки

content = soup.find(id="mw-content-text")

for a in content.find\_all("a", href=True):

href = a['href']

if href.startswith("/wiki/") and not any(prefix in href for prefix in (":", "/File:", "/Категория:", "/Служебная:")):

full = urljoin(BASE, href)

links.add(full)

links = sorted(links)

with open("city\_links.txt", "w", encoding="utf-8") as f:

for u in links:

f.write(u + "\n")

print(f"Found {len(links)} links. Saved to city\_links.txt")

download\_city\_texts:

import requests, time, os, re

from bs4 import BeautifulSoup

from pathlib import Path

from urllib.parse import urlparse, unquote

outdir = Path("cities")

outdir.mkdir(exist\_ok=True)

headers = {"User-Agent":"CorpusBot/1.0 (student@example.com)"}

with open("city\_links.txt", encoding="utf-8") as f:

urls = [line.strip() for line in f if line.strip()]

for i, url in enumerate(urls, start=1):

try:

r = requests.get(url, headers=headers, timeout=30)

r.raise\_for\_status()

soup = BeautifulSoup(r.text, "html.parser")

content = soup.find(id="mw-content-text")

if not content:

content = soup.body

# собираем только параграфы и заголовки (p, h1..h3, li)

parts = []

for tag in content.find\_all(['h1','h2','h3','p','li']):

text = tag.get\_text(separator=" ", strip=True)

if text:

parts.append(text)

text = "\n\n".join(parts)

# очистка лишних пробелов

text = re.sub(r'\n{3,}', '\n\n', text).strip()

# сделаем безопасное имя файла: взять последний сегмент URL

name = unquote(urlparse(url).path.split("/")[-1])

fname = outdir / f"{i:04d}\_{name}.txt"

with open(fname, "w", encoding="utf-8") as out:

out.write(text)

print(f"[{i}/{len(urls)}] Saved {fname} ({len(text)} chars)")

except Exception as e:

print(f"[{i}] ERROR downloading {url}: {e}")

time.sleep(1.2) # пауза между запросами

corpus\_stats\_full:

import re, math, gzip

from collections import Counter

from pathlib import Path

p = Path("corpus\_cities\_raw.txt")

assert p.exists(), "Создайте файл corpus\_cities\_raw.txt (cat cities/\*.txt > ...)"

raw = p.read\_text(encoding='utf-8', errors='ignore')

bytes\_size = p.stat().st\_size

chars = len(raw)

chars\_no\_spaces = len(re.sub(r'\s','', raw))

tokens = re.findall(r"[A-Za-zА-Яа-яЁё0-9']+", raw.lower())

num\_tokens = len(tokens)

vocab = Counter(tokens)

num\_types = len(vocab)

ttr = num\_types / num\_tokens if num\_tokens else 0.0

top50 = vocab.most\_common(50)

freq = Counter(raw)

H = 0.0

for c, cnt in freq.items():

p\_c = cnt / chars

H -= p\_c \* math.log2(p\_c)

gz = gzip.compress(raw.encode('utf-8'), compresslevel=9)

gzip\_size = len(gz)

print("Bytes:", bytes\_size, "(", bytes\_size/1024/1024, "MB )")

print("Chars:", chars, "Chars (no spaces):", chars\_no\_spaces)

print("Tokens:", num\_tokens, "Types:", num\_types, "TTR:", round(ttr,4))

print("Top 50 words:", top50[:50])

print("Shannon entropy (bits/char):", round(H,3))

print("Gzip size (bytes):", gzip\_size, "(", gzip\_size/1024/1024, "MB )")

**Результаты тестирования программы:**

get\_city\_links.py получил все ссылки на статьи городов со страницы «Список городов России» и записал их в city\_links.txt (Рис. 1-2).

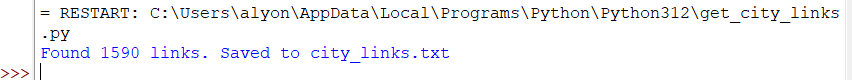


Рис. 1 – результат работы get\_city\_links.py



Рис. 2 – получился файл city\_links.txt

download\_city\_texts.py прочитал city\_links.txt, скачал HTML каждой страницы, извлек основную часть текста и сохраняет в папку cities (Рис. 3-4).

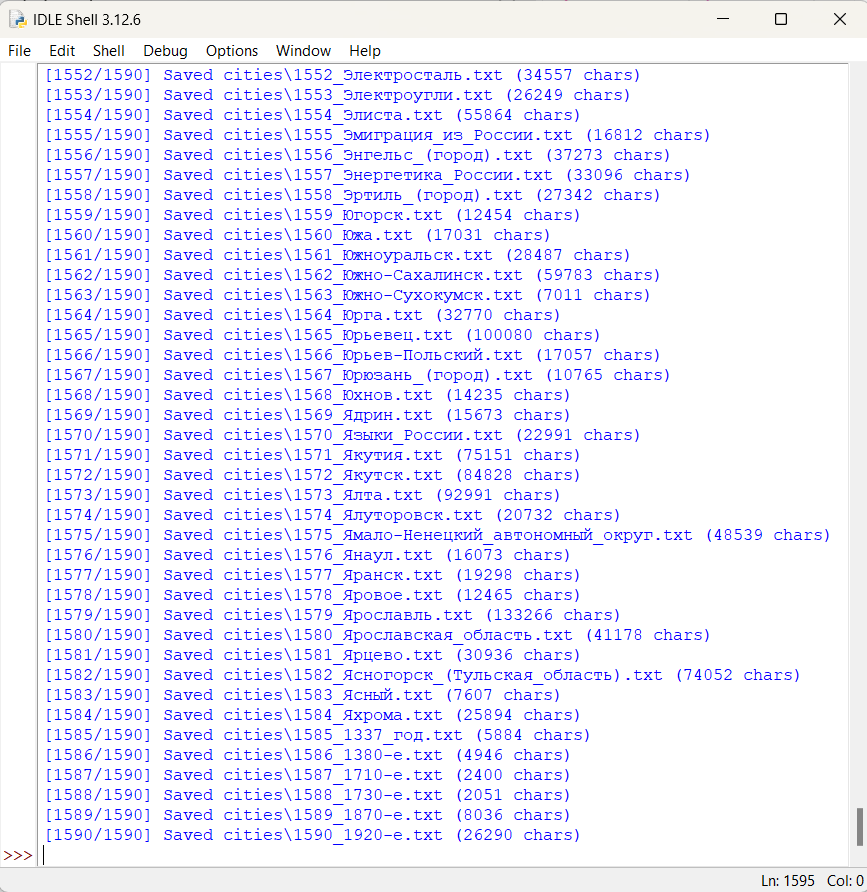


Рис. 3 – результат работы программы download\_city\_texts.py

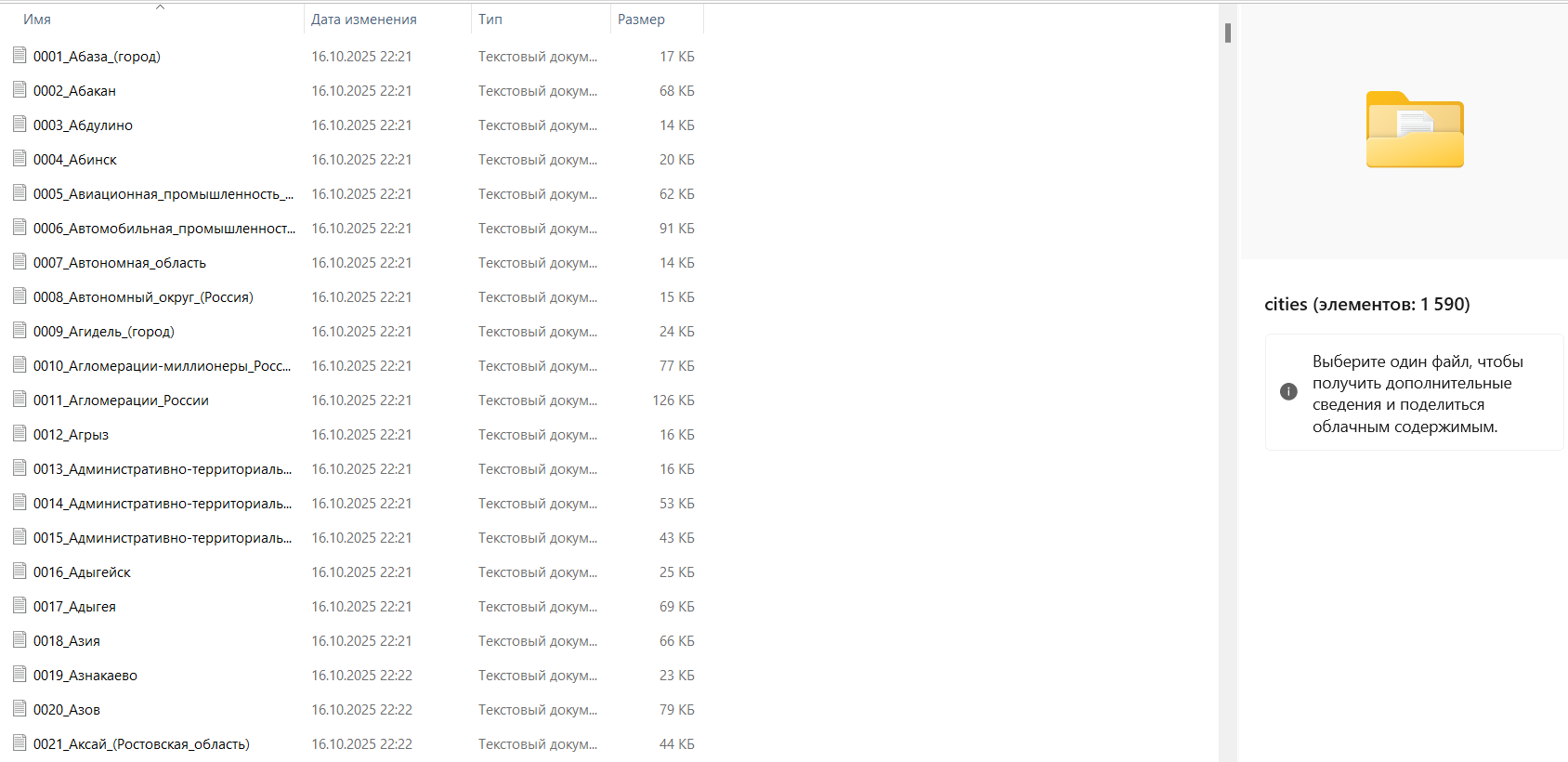


Рис. 4 – образовалась папка cities

Дальше используем Git Bush, чтобы соединить все тексты в один файл и посмотреть размер файла (Рис. 5).

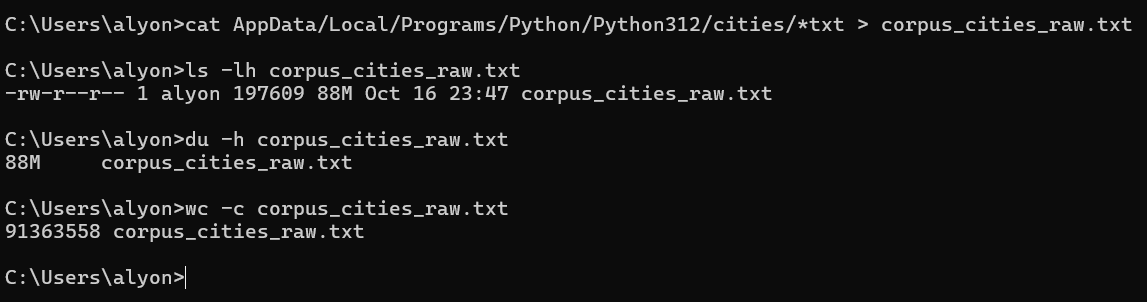


Рис. 5 – соединяем тексты с один и смотрим размер

Результат работы программы corpus\_stats\_full.py (Рис. 6):

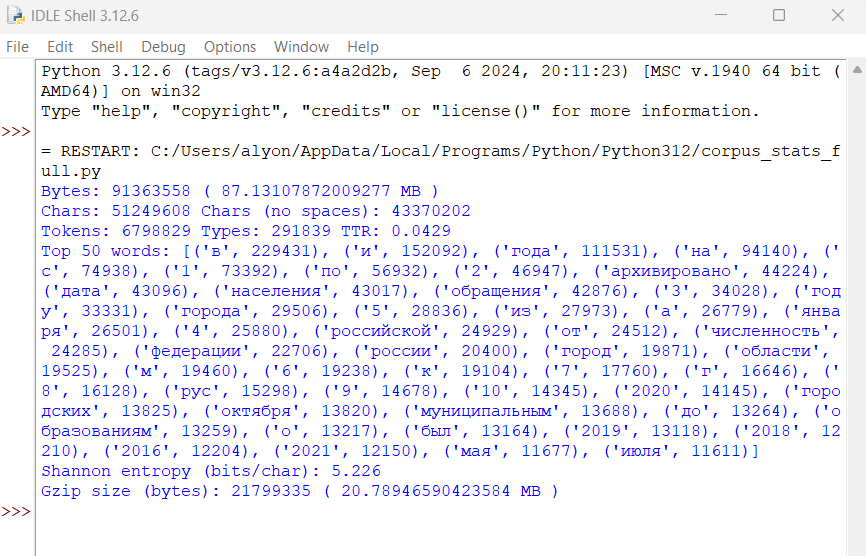


Рис. 6 – результат работы corpus\_stats\_full