## 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 2. Статистические особенности языка

**Постановка задачи:**

Цепи Маркова – это генератор случайных символов, вероятность появления которых зависит от предыдущего символа. Это легко расширяется на общий случай, когда нужно использовать несколько предыдущих символов.

Используйте результат задания 1 и посчитайте вероятности появления:

Буквы после 1, 2, 3, ..., 16 предыдущих букв и в целом.

Учитываем символы русского алфавита (33 буквы, если буква большая, то при подсчёте делаем её маленькой), точку, запятую, пробел, восклицательный и вопросительный знаки.

Код должен быть загружен в публичный репозиторий.

Ответ на задание - ссылка на репозиторий.

# Алгоритм:

**Вход:** файл корпуса corpus\_cities\_raw.txt.  
**Выход:** для каждого k=0..16 файлы outputs/prob\_k.csv и outputs/summary\_k.txt.

1. **Нормализация корпуса**

* Перевести в нижний регистр.
* Разрешённый набор символов: 33 русские буквы (включая ё) + пробел + . , ! ?.
* Все прочие символы заменить на пробел; свернуть подряд идущие пробелы в один; обрезать края.

1. **k = 0 (односимвольное распределение)**

* Посчитать частоты каждого символа .
* Вычислить вероятности .
* Сохранить prob\_k0.csv.

1. **Для каждого k = 1..16**

* Для каждой позиции i (i ≥ k) взять .
* Если все символы контекста и next ∈ разрешённого набора, увеличить счётчик
* Для каждого контекста c вычислить условные вероятности
* Сохранить outputs/prob\_k.csv и outputs/summary\_k.txt (число переходов, число контекстов, условная энтропия H(next|context), топ-контекстов).

1. **Вычисление условной энтропии**
   * + Для каждого контекста :
     + Средняя условная энтропия:  
       , где
2. **Файлы и команды**
   * + Результаты лежат в outputs.

**Листинг программы с описанием системы программирования**

Язык программирования: Python 3.12

Среда выполнения: Windows 11

Среда разработки: IDLE / Visual Studio Code / Git Bash

Установленные библиотеки: стандартные библиотеки Python (collections, math, csv, argparse, pathlib, re, sys)

Назначение программы:

Программа markov\_stats.py вычисляет статистические характеристики текста, основанные на моделях цепей Маркова. Для заданного порядка k строится распределение вероятностей появления символа после k предыдущих символов, а также вычисляется условная энтропия.

import argparse

from collections import Counter, defaultdict

from pathlib import Path

import math, csv, re, sys

RUSSIAN\_LETTERS = list("абвгдеёжзийклмнопрстуфхцчшщъыьэюя")

ALLOWED\_SYMBOLS = set(RUSSIAN\_LETTERS + [' ', '.', ',', '!', '?'])

def normalize\_text(raw):

s = raw.lower()

out = [(ch if ch in ALLOWED\_SYMBOLS else ' ') for ch in s]

res = re.sub(r'\s+', ' ', ''.join(out))

return res.strip()

def conditional\_entropy\_from\_counters(context\_counts):

total = sum(sum(cnt.values()) for cnt in context\_counts.values())

H = 0.0

for cnt in context\_counts.values():

s = sum(cnt.values())

p\_c = s / total

h\_c = -sum((v/s) \* math.log2(v/s) for v in cnt.values())

H += p\_c \* h\_c

return H

def process\_corpus(corpus, start\_k, max\_k, outputs):

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

text = normalize\_text(raw)

n = len(text)

outputs.mkdir(exist\_ok=True)

for k in range(start\_k, max\_k + 1):

prob\_file = outputs / f"prob\_k{k}.csv"

summary\_file = outputs / f"summary\_k{k}.txt"

if prob\_file.exists() and summary\_file.exists():

print(f"k={k} уже обработано, пропуск...")

continue

print(f"Обработка k={k}")

context\_counts = defaultdict(Counter)

for i in range(k, n):

context = text[i-k:i]

next\_ch = text[i]

if all(c in ALLOWED\_SYMBOLS for c in context) and next\_ch in ALLOWED\_SYMBOLS:

context\_counts[context][next\_ch] += 1

with prob\_file.open("w", newline="", encoding="utf-8") as f:

w = csv.writer(f); w.writerow(["context","next","count","probability"])

for c, cnt in context\_counts.items():

total = sum(cnt.values())

for n\_ch, v in cnt.items():

w.writerow([c, n\_ch, v, f"{v/total:.6f}"])

H = conditional\_entropy\_from\_counters(context\_counts)

with summary\_file.open("w", encoding="utf-8") as f:

f.write(f"k={k}\nContexts={len(context\_counts)}\nEntropy={H:.4f}\n")

print(f"Сохранены файлы для k={k}")

if \_\_name\_\_ == "\_\_main\_\_":

p = argparse.ArgumentParser()

p.add\_argument("--corpus", required=True)

p.add\_argument("--start\_k", type=int, default=0)

p.add\_argument("--max\_k", type=int, default=16)

p.add\_argument("--outputs", default="outputs")

args = p.parse\_args()

process\_corpus(Path(args.corpus), args.start\_k, args.max\_k, Path(args.outputs))

Запуск программы (Рис. 1):

python markov\_stats.py --corpus corpus\_cities\_raw.txt

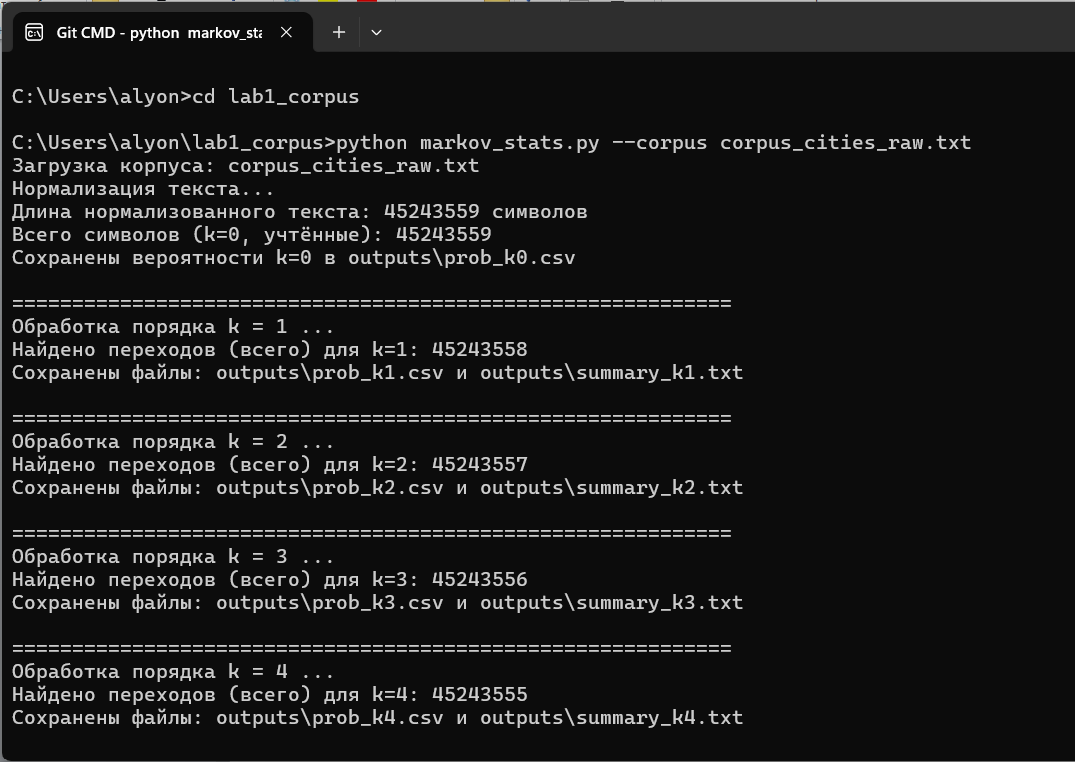


Рис. 1 – запускаем markov\_stats.py в git bush

Образовалась папка outputs с результатами для всех k1,k2,…,k16 (Рис.2)

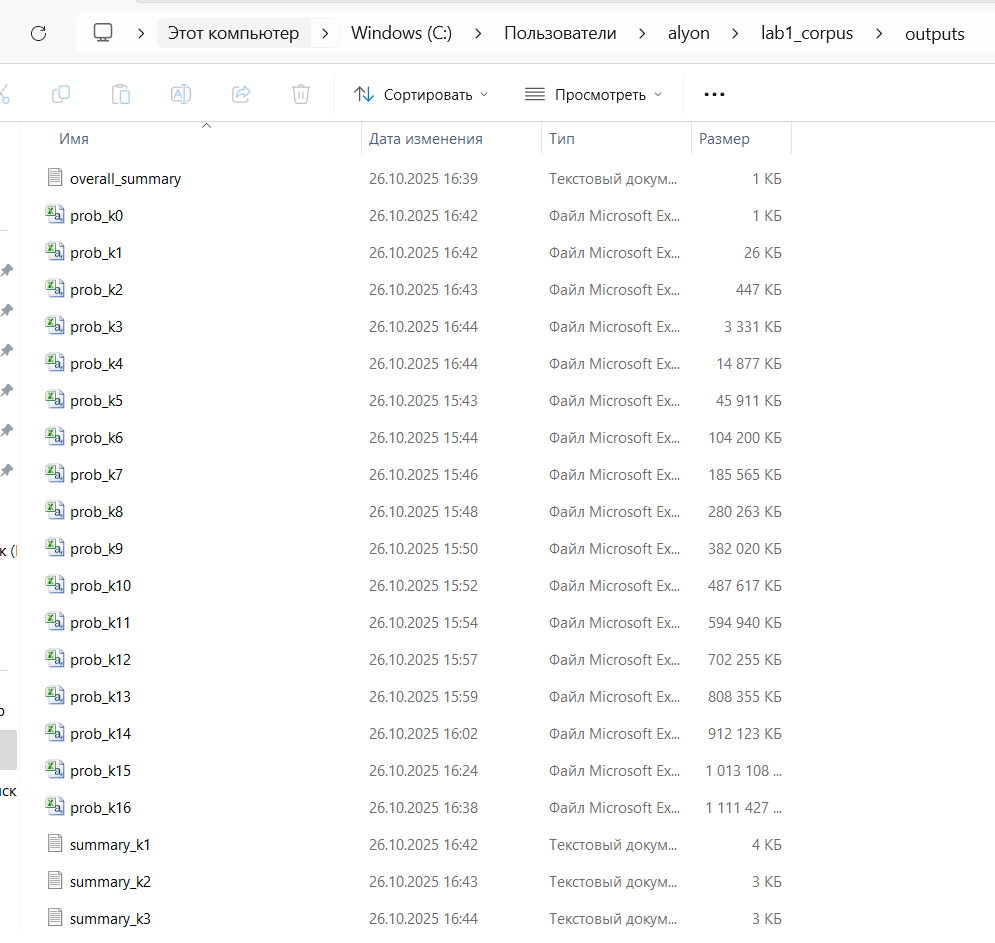


Рис. 2 – результат работы программы – папка outputs

Вот пример результатов для k1 (Рис. 3-4)

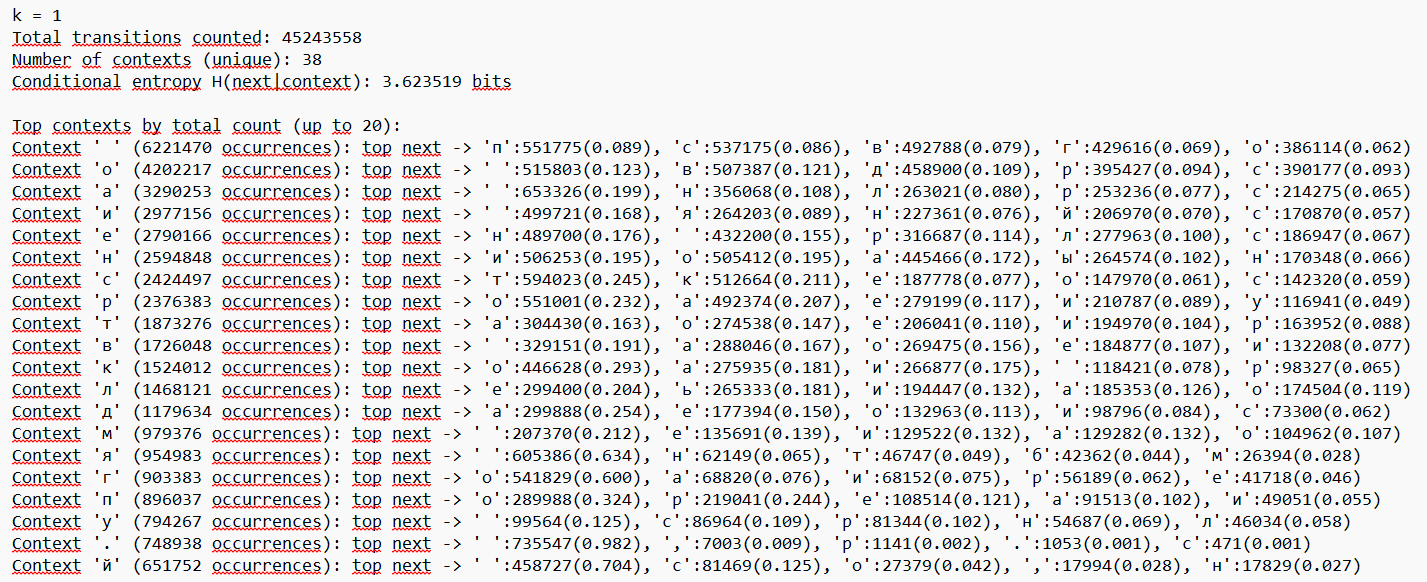


Рис. 3 – скрин файла summary\_k1.txt

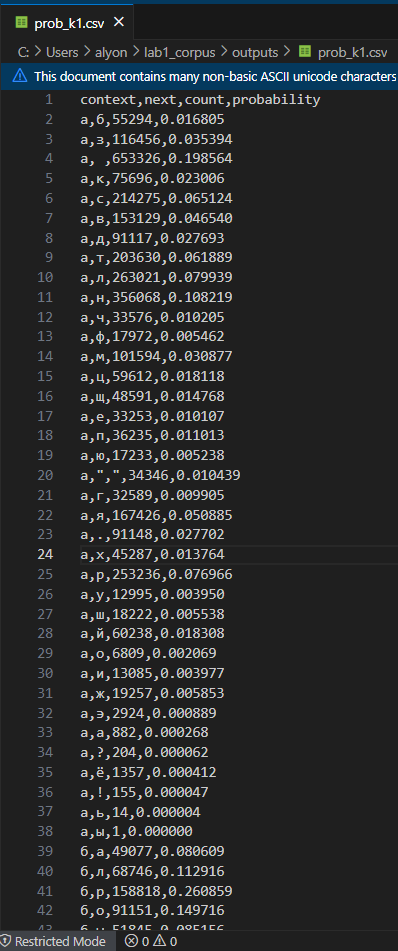


Рис. 4 – скрин файла prob\_k1.csv

**Вывод:** Программа успешно рассчитала статистические характеристики корпуса текстов для порядков k=0…16. С ростом порядка k количество контекстов увеличивается, а условная энтропия снижается, что отражает рост предсказуемости текста. Полученные таблицы и отчёты сохранены в папке outputs.

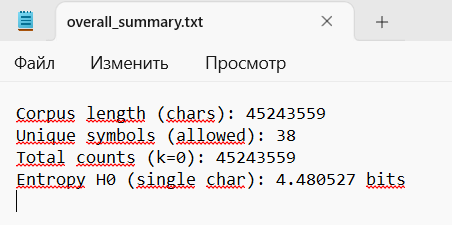


Рис. 5 – полученный файл overall\_summary.txt