Министерство науки и образования Российской Федерации Федеральное автономное образовательное учреждение высшего образования
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
кафедра "Инфокогнитивные технологии"

ЧАСТОТНЫЙ АНАЛИЗ ЕЯ ОПИСАНИЯ ПО

Отчет по выполнению задания 1
курсового проекта
студент группы 221-321
Переверзев Иван Дмитриевич.

Преподаватели:
к.т.н., профессор Юрий Николаевич Филиппович
к.ф.н., доцент Олеся Анатольевна Змазнева
ассистент Владислав Алексеевич Речинский.

Москва, 2025г.

ОТЧЕТ ПО ВЫПОЛНЕНИЮ ЗАДАНИЯ 1

Формулировка задания

Задание 1

При проведении частотного анализа ЕЯ описания ПО необходимо выполнить следующие работы:

- построить частотные словники;
- вычислить основные частотные характеристики и представить их в таблицах:
    - ранг слова – порядковый номер этого слова в частотном словаре, в котором все лексические единицы упорядочены по частоте;
    - абсолютная частота слова – общее количество словарных единиц в корпусе текстов;
    - относительная частота слова – представляет долю данного слова от общего числа всех терминов в тексте;
    - нормализированный ранг слова – логарифм от ранга слова;
    - нормализированная относительная частота слова – логарифм от относительной частоты слова.
- построить графики ступенчатой функции распределения частот, используя ранг слова (ось Y) и абсолютную частоту слова (ось X);
- построить графики распределения частот слов, используя нормализированный ранг слова (ось X) и нормализированную относительную частоту слова (ось Y).

Частотный словник (или частотный словарь) – представляет из себя набор из слов, принадлежащий отдельно взятому тексту, вместе с набором информации о частоте этих слов.

Данный этап необходим для сортировки, информации, которая будет использоваться при построении графов, отображающих структуру выбранной студентом научной документации.

В качестве результатов выполнения задания необходимо прикрепить отчет, в котором описывается последовательность действий по выполнению задания, а также документы (файлы словарей) где задание реализовано.

В случае, если исследуется набор текстов – необходимо составить общий частотный словник и провести частотный анализ текстов и сделать тоже самое для каждого из текстов отдельно.

Дополнительное . задание 1

Составить словарь лемм. Лемма в лингвистике — это начальная, словарная форма слова. Она служит основой для всех грамматических вариаций слова и помогает «узнать» слово в разных формах. Для существительных и прилагательных в русском языке лемма — это именительный падеж единственного числа, например, «кот» или «красный». Для глаголов и глагольных форм — это инфинитив, то есть «бежать» или «читать».

Лемма используется в корпусной лингвистике и автоматической обработке естественного языка. Она позволяет более эффективно анализировать тексты, например, «понимая», что слова «котами» и «коту» — это разные формы одного и того же слова, «кот».

Структура словаря должна иметь следующую структуру:

<Лемма>
(Словоформа1)
(Словоформа2)
(Словоформа3)
…

(СловоформаN)

Где лемма – слово в основной форме; СловоформаN – список словоформ, связанных с этой леммой.

Кроме этого, необходимо провести удаление стоп-слов в тексте, после чего для полученного списка лемм сделать следующее:

- построить частотные словники;
- вычислить основные частотные характеристики и представить их в таблицах:
  - ранг слова – порядковый номер этого слова в частотном словаре, в котором все лексические единицы упорядочены по частоте;
  - абсолютная частота слова – общее количество словарных единиц в корпусе текстов;
  - относительная частота слова – представляет долю данного слова от общего числа всех терминов в тексте;
  - нормализированный ранг слова – логарифм от ранга слова;
  - нормализированная относительная частота слова – логарифм от относительной частоты слова.
- построить графики ступенчатой функции распределения частот, используя ранг слова (ось Y) и абсолютную частоту слова (ось X);
- построить графики распределения частот слов, используя нормализированный ранг слова (ось X) и нормализированную относительную частоту слова (ось Y).

Дополнительное . задание 2

Для слов на английском языке должны быть представлены переводы (количество вариантов перевода не более трех).

Дополнительное . задание 3

Для поэтических текстов должны быть составлены обратные словники.

В обратном словнике словоформы отсортированы в алфавитном порядке с учетом обратного чтения, т.е. не по начальным буквам, а по конечным.

Ход работы

После получения задания для курсовой работы на площадке github был создан репозиторий, в котором будут хранится все выполненные задания. Ссылка на репозиторий была передана одному из преподавателей по данной дисциплине.

Важные уточнения

Перед описанием основных действий при выполнении задания 1 стоит уточнить следующие моменты:

- Все программы были написаны на языке программирования Python версии 3.11.7.
- Все программы были написаны на ОС Linux, поэтому запуск на других платформах может быть затруднен, но возможен.
- Для хранения частотных словников и словаря лемм с их словоформами был выбран формат файлов JSON, так как он просто как в записи, так и в чтении программами, написанными на python.
- В ходе выполнения задания предполагалось использование орфографической проверки анализируемых слов с помощью библиотеки PyEnchant. Однако, словарь данной библиотеки оказался слишком ограниченным, из-за чего многие слова (например, аббревиатуры) перестали учитываться в анализе, хоть и написаны правильно. Поэтому на данном этапе было принято решение отключить данную проверку.

Начальная структура проекта

После создания репозитория последовало определение файловой структуры всего проекта. В конечном итоге было принято решение использовать следующую структуру папок и файлов:

- ./taskN — каталоги для хранения выполненных заданий курсовой работы (N - номер задания). Для получения более подробной информации о каждом задании мы можете перейти в соответствующие им директории и прочитать хранящиеся в них readme файлы.
- ./texts — каталог для хранения текстов, которые необходимо проанализировать, а также для программы конвертации pdf-файлов в удобочитаемые для других программ txt-файлы.
- ./requirements.txt — файл, в котором записаны все необходимые для запуска программ библиотеки.
- Требования.doc — текстовый документ с описанием всех заданий курсовой работы. Был взят скачан из курса ЛМС.

После определения файловой структуры проекта были скачаны тексты, которые в будущем предстояло анализировать, в соответствии с определенным преподавателями вариантом — 21 вариантом. Все тексты, кроме поэтических, были даны в формате PDF документов. Все поэтические тексты, а именно стихи Марины Цветаевой «В огромном городе моем ночь...» и «Уж сколько их упало в эту бездну ...», были скачаны из сети Интернет сразу в удобочитаемом для написанных позже программ текстом формате (.txt).

Входные данные и их адаптация

Далее, после скачивания анализируемых материалов, было принято решение написать небольшую программу, которая помогала бы считывать текст, записанный в PDF файлах, обрабатывать его и записывать в текстовый файл (.txt). В итоге был написан следующий программный код:

```
# texts/convert_pdf_to_txt.py

import pdfplumber
import re
import os


def clean_text(text):
    """Очищает текст от переносов слов и лишних разрывов строк."""
    # Соединяем слова, разорванные дефисом на переносе строки (сло-\nво → слово)
    text = re.sub(r"(\w+)-\n(\w+)", r"\1\2", text)

    # Убираем лишние разрывы строк, заменяя их пробелами (но не между пунктами списка и
заголовками)
    text = re.sub(r"(?<!\n)\n(?!\n)", " ", text)

    return text


def extract_text_from_pdf(pdf_path):
    full_text = []
    previous_page_text = ""

    with pdfplumber.open(pdf_path) as pdf:
        for page in pdf.pages:
            # Проверяем, есть ли текст на странице
            current_page_text = page.extract_text()
            if not current_page_text:
                continue  # Пропускаем пустые страницы

            cleaned_text = clean_text(current_page_text)

            # Обрабатываем переносы слов между страницами
            if previous_page_text:
                match = re.search(r"(\w+)-$", previous_page_text)
                if match:
                    unfinished_word = match.group(1)
                    cleaned_text = re.sub(
                        rf"^{unfinished_word}(\w+)", rf"\1", cleaned_text
                    )

            full_text.append(cleaned_text)
            previous_page_text = current_page_text

    return "\n".join(full_text)


def save_text_to_file(text, output_path):
    with open(output_path, "w", encoding="utf-8") as f:
        f.write(text)


def process_pdfs(input_folder, output_folder):
    for filename in os.listdir(input_folder):
        if filename.endswith(".pdf"):
            pdf_path = os.path.join(input_folder, filename)
            txt_path = os.path.join(output_folder, filename.replace(".pdf", ".txt"))

            parsed_text = extract_text_from_pdf(pdf_path)
            save_text_to_file(parsed_text, txt_path)

            print(f"Обработанный текст {filename} сохранён в {txt_path}")
```

```
# Использование:
input_folder = "./pdf/"  # Укажи путь к папке с PDF-файлами
output_folder = "./txt/"  # Укажи путь к папке для сохранения текстовых файлов

# Создаём папку для вывода, если она не существует
if not os.path.exists(output_folder):
    os.makedirs(output_folder)

process_pdfs(input_folder, output_folder)
```

Данный код приемлемо справился с задачей считывания текста из PDF файла, его очистки от лишних символов и записи в текстовый файл. Однако была выявлена некоторая особенность предоставленных PDF документов, которая негативно повлияла на точность считывания текста. А именно запись текста колонками (Рисунок 1). Из-за данной особенности в конечный вариант текста для обработки попадали некорректные слова (Рисунок 2). Однако данные слова составляли незначительную долю от всего текста, который по большей части был обработан правильно.

А.А. Павлова AnniaPavlova@yandex.ru

МГТУ им. Н.Э. Баумана, Москва, Российская Федерация

Аннотация

Рассмотрены USB-накопители, в частности, внутренние компоненты USB-устройства, современные комплексы, используемые в рамках производства судебной компьютерно-технической экспертизы при работе с USB-накопителями информации, а именно: аппаратно-программный комплекс PC-3000 flash, позволяющий восстановить данные с поврежденных USB-устройств, и программный комплекс Encase Forensic, который применяется для поиска, анализа и восстановления данных с USB-устройств.

Ключевые слова

USB-накопитель, судебная компьютерно-техническая экспертиза, программные средства, поврежденные устройства, восстановление данных, поиск данных

Поступила в редакцию 06.10.2017

Рисунок 1 — Пример записи текста колонками в предоставленных PDF файлах

1 УДК 343.148.6 DOI: 10.18698/2541-8009-2017-12-215 ИССЛЕДОВАНИЕ USB-НАКОПИТЕЛЕЙ В РАБОТЕ СУДЕБНОГО КОМПЬЮТЕРНО-ТЕХНИЧЕСКОГО ЭКСПЕРТА А.А. Павлова AnniaPavlova@yandex.ru МГТУ им. Н.Э. Баумана, Москва, Российская Федерация Аннотация Ключевые слова Рассмотрены USB-накопители, в частности, внут- USB-накопитель, судебная комренние компоненты USB-устройства, современные пьютерно-техническая эксперкомплексы, используемые в рамках производства тиза, программные средства, судебной компьютерно-технической экспертизы при поврежденные устройства, восработе с USB-накопителями информации, а именно: становление данных, поиск данаппаратно-программный комплекс PC-3000 flash, ных позволяющий восстановить данные с поврежденных USB-устройств, и программный комплекс Encase Forensic, который применяется для поиска, анализа Поступила в редакцию 06.10.2017 и восстановления данных с USB-устройств. © МГТУ им. Н.Э. Баумана, 2017 Введение. Актуальность исследования USB-накопителей обусловлена их широким использованием в настоящее время, поскольку это наиболее удобные и практически незаменимые переносные устройства для хранения данных, которые способны сочетать в себе такие свойства, как компактность и возможность хранения большого объема информации. В связи с этим данные устройства нередко становятся объектами исследования при проведении судебной компьютерно-технической экспертизы, поскольку могут содержать криминалистически важную информацию. USB-накопитель — это запоминающее устройство, используемое для хранения и записи данных с интегрированным интерфейсом USB [1]. Все данные записываются и хранятся в микросхеме памяти, для записи и чтения которых необходимо подключить USB-накопитель в любое считывающее устройство, где предусмотрен интерфейс USB [2]. Безусловно, на сегодняшний день USB-накопитель является одним из наиболее компактных и удобных запоминающих устройств, в котором отсутствуют движущиеся элементы, как в оптических или в жестких магнитных дисках. Это — набор микросхем, в чипах которых способна храниться цифровая информация, поэтому нет необходимости использовать батарейки или аккумуляторы при работе с устройством. Кроме того, существует множество производителей данных устройств, выпускающих USB-накопители с различным объемом памяти (до 1 Тбайт) и различной пропускной способностью (USB 2.0, 3.0, 3.1 и др.). Структура USB-накопителя. Устройство USB-накопителя может включать в себя до восьми составляющих элементов (рис. 1) [3]. Политехнический молодежный журнал. 2017. № 12 1

Рисунок 2 — Пример некорректного считанного текста из PDF файла

Структура реализации задания 1

После написания кода для конвертации текста из PDF в TXT началась работа над основной частью задания 1. Для этого в корне проекта был создан каталог «task1», внутри которого находятся следующие файлы и директории:

- ./base.py — реализация основной части задания
- ./lemmas.py - 1 дополнительное задание
- ./translate.py - 2 дополнительное задание
- ./poetry.py - 3 дополнительное задание
- ./test.py - файл для тестирования отдельных функций
- ./modules - вспомогательные модули, функции из которых вызываются в основных файлах
- ./results - каталог для хранения результатов выполнения программ

Для хранения результатов проекта была определена следующая структура каталога result:

```
./results
├── graphics — графики
│   ├── base — графики для основного задания
│   │   ├── common — для обычных рангов и частот
│   │   └── normalize — для нормализированных рангов и частот
│   └── lemmas — графики для 1 дополнительного задания (лемм)
│       ├── common — для обычных рангов и частот
│       └── normalize — для нормализированных рангов и частот
├── lemmas — словари лемм и их словоформ
├── poetry — обратные словники для поэтических текстов
├── slovniks — частотные словники
│   ├── base — для базового задания
│   └── lemmas — для 1 доп. задания (лемм)
├── tables — таблицы с частотными характеристиками
│   ├── base — для основного задания (всех словоформ)
│   └── lemmas - для 1 доп. задания (лемм)
└── translate — словари с переводами английских слов
```

Вся основная логика хранится в папке «modules», в которой хранятся следующие файлы:

```
./modules
├── graphics.py — для построения графиков
├── lang_utils.py — для работы с языками (определение языка, орф. проверка)
├── lemmas.py — для работы с леммами
├── poetry.py — для создания обратных словников
├── slovnik.py — для создания частотных словников из TXT
├── stop_words.py — стоп-слова для удаления из текста (пока не используется)
├── tables.py — для создания таблиц с частотными характеристиками
├── translate.py — для перевода английских слов
└── utils.py — переиспользуемые вспомогательные функции
```

В свою очередь, файла наподобие «./translate.py» нужны только для того, чтобы применять функции из каталога «modules» сразу к нескольким файлам и представляют из себя следующий код:

```
# task1/translate.py

import os
from modules.translate import generate_translation_dict

txt_dir = "../texts/txt/"
result_dir = "./results"
```

```
translate_dir = result_dir + "/translate/"


def process_files(folder_path):
    """Принимает на вход папку каталог, в котором хранятся файлы для обработки,
    считывает из них все слова и создает словари с переводом английских слов.
    """

    # Проверка, существует ли папка
    if not os.path.exists(folder_path):
        print(f"Папка '{folder_path}' не найдена.")
        return

    # Получение списка всех файлов в папке
    txt_files = [f for f in os.listdir(folder_path) if f.endswith(".txt")]

    # Проверка наличия файлов формата .txt
    if not txt_files:
        print("В папке нет файлов формата .txt.")
        return

    # Обработка каждого файла
    for txt_file in txt_files:
        # Получение полного пути к файлу
        full_path = os.path.join(folder_path, txt_file)
        translate_file = os.path.join(translate_dir, txt_file.replace(".txt", ".json"))
        generate_translation_dict(full_path, translate_file)

    print("Перевод файлов завершен.")


confirm = input(
    "Осторожно: данный скрипт обращается к внешнему API, из-за чего может работать очень долго.\n"
    "Если действительно хотите заново перевести все файлы, то напишите слово 'yes': "
)
if confirm == "yes":
    process_files(txt_dir)
else:
    print("Перевод слов в текстах отменен")
```

Данный пример типичен для всех программных файлов, находящихся на том же уровне вложенности. Поэтому их код не будет представлен в данном отчете.

Дополнительные удобства

Также важно отметить, что во время выполнения задания 1 активно применялась практика использования Make-файлов для более удобного запуска написанных программ. На данный момент в make-файле доступны следующие команды:

- make base — Запуск кода для выполнения основного задания, а также создание каталогов для хранения результатов.
- make lemmas — Запуск кода для выполнения первого дополнительного задания, а также создание каталогов для хранения результатов.
- make translate — Запуск кода для выполнения второго дополнительного задания, а также создание каталогов для хранения результатов.
- make poetry — Запуск кода для выполнения третьего дополнительного задания, а также создание каталогов для хранения результатов.
- make run — Запуск всех перечисленных выше программ.
- make clean — Очистка удаление всех созданных ранее результатов.

Результаты работы

Реализованный функционал

В результате выполнения задания 1 был написан программный код, который способен выполнять следующие задачи:

- Построение частотных словников.
- Вычисление основных частотных характеристик и представление их в таблицах:
  - Ранг слова
  - Абсолютная частота слова
  - Относительная частота слова
  - Нормализированный ранг слова
  - Нормализированная относительная частота
- Построение графика ступенчатой функции распределения частот.
- Построение графика распределения частот слов.
- Создание словаря лемм.
- Перевод английских слов на русский язык.
- Создание обратных словников для поэтических текстов.

Также в репозиторий проекта были добавлены README файлы, которые подробно описывают его структуру, а также содержат инструкции для запуска кода на чужом компьютере.

Демонстрация результатов работы

После выполнения всех написанных программных файлов с помощью команды «make run» в папку *results* были сохранены результаты их выполнения, демонстрация которых представлена ниже.

Небольшая часть частотного словника, составленного для одного из текстов:


```
{
    "удк": 1,
    "3431486": 1,
    "doi": 1,
    "10186982541-8009-2017-12-215": 1,
    "исследование": 8,
    "usb-накопителей": 12,
    "работе": 10,
    "судебного": 8,
    "компьютерно-технического": 8,
    "эксперта": 11,
    "аа": 8,
    "павлова": 9,
    "anniapavlovayandexru": 2,
    "мгту": 4,
    "нэ": 4,
    "баумана": 4,
    "москва": 5,
    "российская": 3,
    "федерация": 3,
    "аннотация": 1,
    "ключевые": 1,
    "слова": 1,
    "рассмотрены": 1,
    "usb-накопители": 3,
    "частности": 1,
    "внут-": 1,
    "usb-накопитель": 10,
    "судебная": 2,
    "комренние": 1,
```

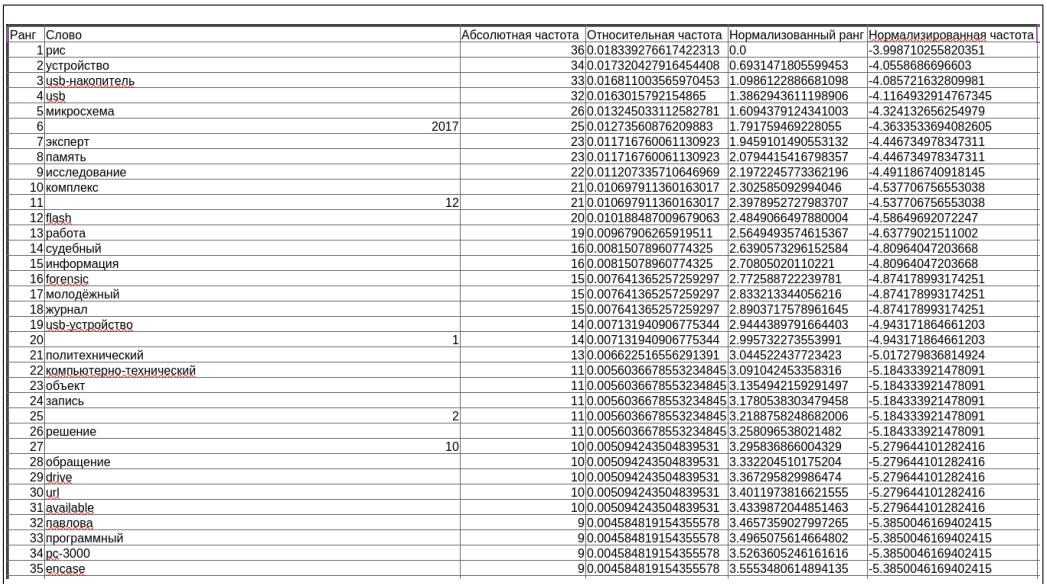
```
    "компоненты": 1,
    "usb-устройства": 6
}
```

Небольшая часть словаря лемм и их словоформ, составленного для одного из текстов:

```
{
    "исследование": [
        "исследование",
        "исследования",
        "исследованию",
        "исследовании"
    ],
    "usb-накопитель": [
        "usb-накопителей",
        "usb-накопитель",
        "usb-накопителями",
        "usb-накопителя",
        "usb-накопителях"
    ],
    "работа": [
        "работе",
        "работу",
        "работы"
    ],
    "судебный": [
        "судебного",
        "судебная",
        "судебной",
        "судебным"
    ],
    "компьютерно-технический": [
        "компьютерно-технического",
        "компьютерно-технической",
        "компьютерно-технический"
    ],
    "эксперт": [
        "эксперта",
        "экспертом",
        "эксперт",
        "эксперту"
    ]
}
```

Ранг	Слово	Абсолютная частота	Относительная частота	Нормализованный ранг	Нормализированная частота
1	рис	36	0.018339276617422313	0.0	-3.998710255820351
2	2017	25	0.01273560876209883	0.6931471805599453	-4.3633533694082605
3	usb	24	0.012226184411614875	1.0986122886681098	-4.404175363928515
4	12	21	0.010697911360163017	1.3862943611198906	-4.537706756553038
5	памяти	19	0.00967906265919511	1.6094379124341003	-4.63779021511002
6	flash	18	0.009169638308711156	1.791759469228055	-4.691857436380296
7	forensic	15	0.007641365257259297	1.9459101490553132	-4.874178993174251
8	молодежный	15	0.007641365257259297	2.0794415416798357	-4.874178993174251
9	журнал	15	0.007641365257259297	2.1972245773362196	-4.874178993174251
10	1	14	0.007131940906775344	2.302585092994046	-4.943171864661203
11	политехнический	13	0.006622516556291391	2.3978952727983707	-5.017279836814924
12	usb-накопителей	12	0.006113092205807438	2.4849066497880004	-5.097322544488461
13	устройства	12	0.006113092205807438	2.5649493574615367	-5.097322544488461
14	исследования	12	0.006113092205807438	2.639057329615258	-5.097322544488461
15	комплекса	12	0.006113092205807438	2.70805020110221	-5.097322544488461
16	эксперта	11	0.005603667855323484	2.772588722239781	-5.184333921478091
17	устройство	11	0.005603667855323484	2.833213344056216	-5.184333921478091
18	2	11	0.005603667855323484	2.8903717578961645	-5.184333921478091
19	работе	10	0.005094243504839531	2.9444389791664403	-5.279644101282416
20	usb-накопитель	10	0.005094243504839531	2.995732273553991	-5.279644101282416
21	записи	10	0.005094243504839531	3.044522437723423	-5.279644101282416
22	10	10	0.005094243504839531	3.091042453358316	-5.279644101282416
23	url	10	0.005094243504839531	3.1354942159291497	-5.279644101282416
24	available	10	0.005094243504839531	3.1780538303479458	-5.279644101282416
25	павлова	9	0.004584819154355578	3.2188758248682006	-5.3850046169402415
26	информации	9	0.004584819154355578	3.258096538021482	-5.3850046169402415
27	комплекс	9	0.004584819154355578	3.295836866004329	-5.3850046169402415
28	pc-3000	9	0.004584819154355578	3.332204510175204	-5.3850046169402415
29	encase	9	0.004584819154355578	3.367295829986474	-5.3850046169402415
30	микросхемы	9	0.004584819154355578	3.4011973816621555	-5.3850046169402415
31	эксперт	9	0.004584819154355578	3.4339872044851463	-5.3850046169402415
32	дата	9	0.004584819154355578	3.4657359027997265	-5.3850046169402415
33	обращения	9	0.004584819154355578	3.4965075614664802	-5.3850046169402415
34	moscow	9	0.004584819154355578	3.5263605246161616	-5.3850046169402415
35	accessed	9	0.004584819154355578	3.5553480614894135	-5.3850046169402415

Рисунок 3 — Небольшая часть таблицы с частотными характеристиками слов одного из текстов



Ранг	Слово	Абсолютная частота	Относительная частота	Нормализованный ранг	Нормализированная частота
1	рис	36	0.018339276617422313	0.0	-3.998710255820351
2	устройство	34	0.017320427916454408	0.6931471805599453	-4.0558686696603
3	usb-накопитель	33	0.016811003565970453	1.0986122886681098	-4.085721632809981
4	usb	32	0.01630157921548465	1.3862943611198906	-4.1164932914767345
5	микросхема	26	0.013245033112582781	1.6094379124341003	-4.324132656254979
6	2017	25	0.01273560876209883	1.791759469228055	-4.3633533694082605
7	эксперт	23	0.011716760061130923	1.9459101490553132	-4.446734978347311
8	память	23	0.011716760061130923	2.0794415416798357	-4.446734978347311
9	исследование	22	0.011207335710646969	2.1972245773362196	-4.491186740918145
10	комплекс	21	0.010697911360163017	2.302585092994046	-4.537706756553038
11	12	21	0.010697911360163017	2.3978952727983707	-4.537706756553038
12	flash	20	0.010188487009679063	2.4849066497880004	-4.58649692072247
13	работа	19	0.00967906265919511	2.5649493574615367	-4.63779021511002
14	судебный	16	0.00815078960774325	2.639057329615258	-4.809640472036668
15	информация	16	0.00815078960774325	2.70805020110221	-4.809640472036668
16	forensic	15	0.007641365257259297	2.772588722239781	-4.874178993174251
17	молодёжный	15	0.007641365257259297	2.833213344056216	-4.874178993174251
18	журнал	15	0.007641365257259297	2.8903717578961645	-4.874178993174251
19	usb-устройство	14	0.007131940906775344	2.9444389791664403	-4.943171864661203
20	1	14	0.007131940906775344	2.995732273553991	-4.943171864661203
21	политехнический	13	0.006622516556291391	3.044522437723423	-5.017279836814924
22	компьютерно-технический	11	0.005603667855323484	3.091042453358316	-5.184333921478091
23	объект	11	0.005603667855323484	3.1354942159291497	-5.184333921478091
24	запись	11	0.005603667855323484	3.1780538303479458	-5.184333921478091
25	2	11	0.005603667855323484	3.2188758248682006	-5.184333921478091
26	решение	11	0.005603667855323484	3.258096538021482	-5.184333921478091
27	10	10	0.005094243504839531	3.295836866004329	-5.279644101282416
28	обращение	10	0.005094243504839531	3.332204510175204	-5.279644101282416
29	drive	10	0.005094243504839531	3.367295829986474	-5.279644101282416
30	url	10	0.005094243504839531	3.4011973816621555	-5.279644101282416
31	available	10	0.005094243504839531	3.4339872044851463	-5.279644101282416
32	павлова	9	0.004584819154355578	3.4657359027997265	-5.3850046169402415
33	программный	9	0.004584819154355578	3.4965075614664802	-5.3850046169402415
34	pc-3000	9	0.004584819154355578	3.5263605246161616	-5.3850046169402415
35	encase	9	0.004584819154355578	3.5553480614894135	-5.3850046169402415

Рисунок 4 — Небольшая часть таблицы с частотными характеристиками слов одного из текстов

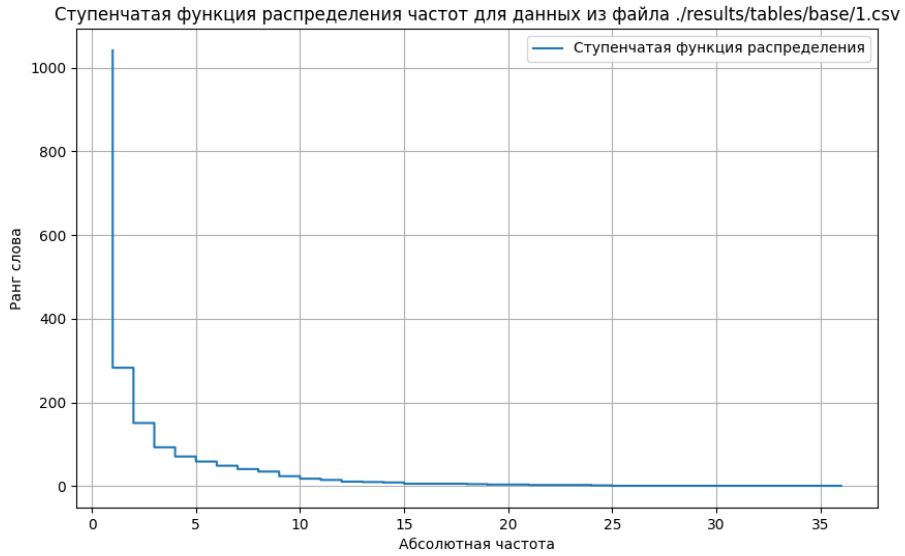


Рисунок 5 — Пример графика распределения частот слов в соответствии с их рангами

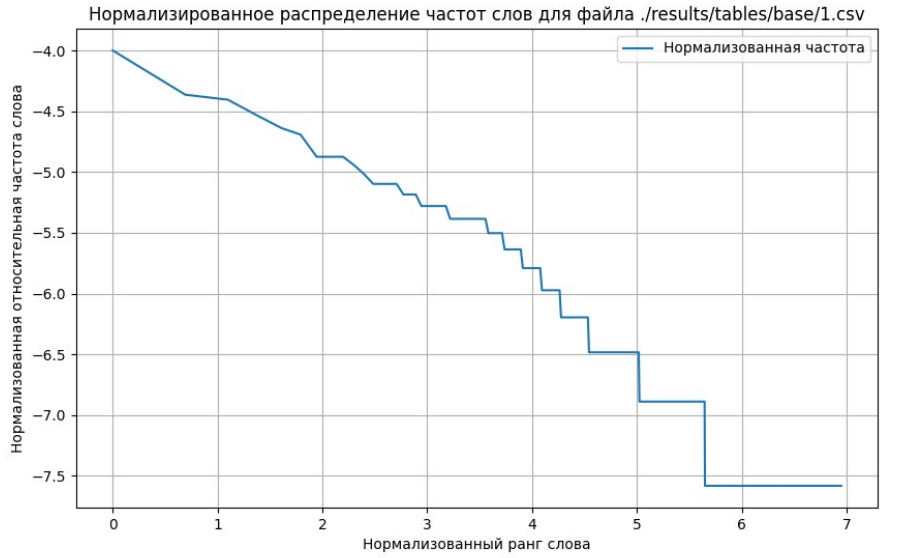


Рисунок 6 — Пример графика распределения нормализированных частот слов в соответствии с их нормализированными рангами

Рисунок 7 — Пример графика распределения частот лемм слов в соответствии с их рангами

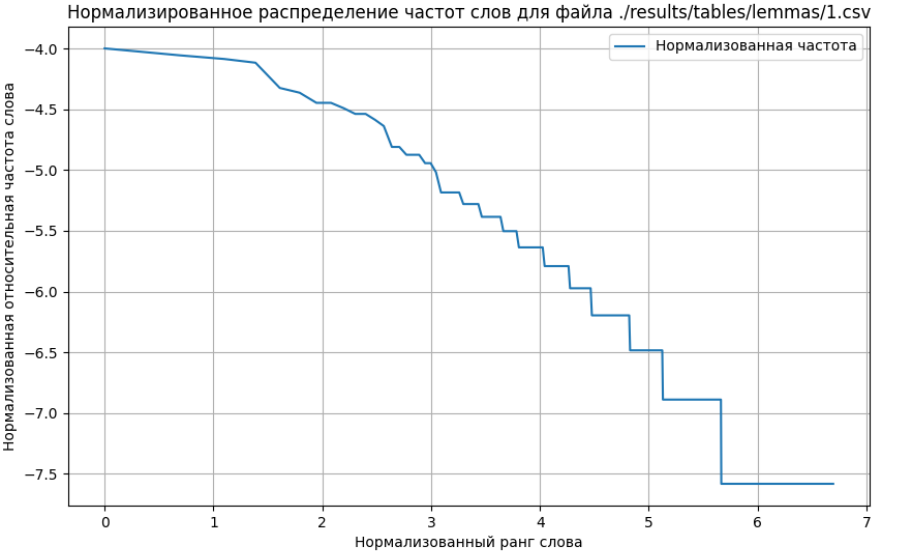


Рисунок 8 — Пример графика распределения нормализированных частот лемм слов в соответствии с их нормализированными рангами

Небольшая часть словаря переведенных английских слов для одного из текстов:

```
{
    "study": [
        "изучать"
    ],
    "drives": [
        "приводы"
    ],
    "work": [
```

```
        "работа"
    ],
    "computer": [
        "компьютер"
    ],
    "expert": [
        "эксперт"
    ],
    "aa": [
        "aa"
    ],
    "pavlova": [
        "Павлова"
    ],
    "moscow": [
        "Москва"
    ],
    "state": [
        "состояние"
    ],
    "technical": [
        "технический"
    ]
}
```

Небольшая часть обратного словника для поэтических текстов:

```
дрова
листика
музыка
запомнила
дома
жена
шаг
вслед
обид
вид
городе
чужие
тонкие
прощение
руке
селе
земле
мине
камине
окне
башне
любите
освободите
поймите
послушайте
глаз
уз
любви
```