Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Кафедра интеллектуальных информационных технологий

ОТЧЁТ

по курсу «Естественно-языковой интерфейс интеллектуальных систем»

Лабораторная работа №2

«Построение и использование корпусов текстов естественного языка»

Выполнил:

студент гр. 221701

Телица И.Д.

Проверил:

Крапивин Ю.Б.

Минск, 2025

# Цели работы

**1.** Изучить принципы построения корпусов текстов, виды разметки и способы аннотирования, инструменты работы с корпусами текстов.

**2.** Построить корпус текстов и разработать корпусный менеджер.

# Задание

1. Сформировать электронный корпус текстов предметной области «спорт» (язык текста — русский).

2. Используя результаты лабораторной работы №1, разработать корпусный менеджер, обеспечивающий базовую функциональность работы с созданным корпусом текстов.

# Используемые библиотеки

**1. os:** Стандартная библиотека Python для работы с файловой системой.

**2. json:** Стандартная библиотека Python для работы с JSON-данными.

**3. collections:** Стандартная библиотека Python для специализированных контейнеров (defaultdict, Counter).

**4. python-docx:** Библиотека для чтения DOCX-файлов.

**5. pdfplumber:** Библиотека для извлечения текста из PDF-файлов.

**6. nltk:** Библиотека для обработки естественного языка, используется для токенизации.

**7. pymorphy2:** Библиотека для морфологического анализа русского языка.

**8. tkinter:** Стандартная библиотека Python для создания GUI.

# Интерфейс

Интерфейс включает:

* • **Панель управления:** Кнопки «Загрузить текст», «Сохранить корпус», «Показать статистику».
* • **Панель поиска:** Поле ввода, радиокнопки «Слово»/«Лемма», кнопка «Искать».
* • **Область вывода:** Текстовое поле для результатов поиска и статистики.

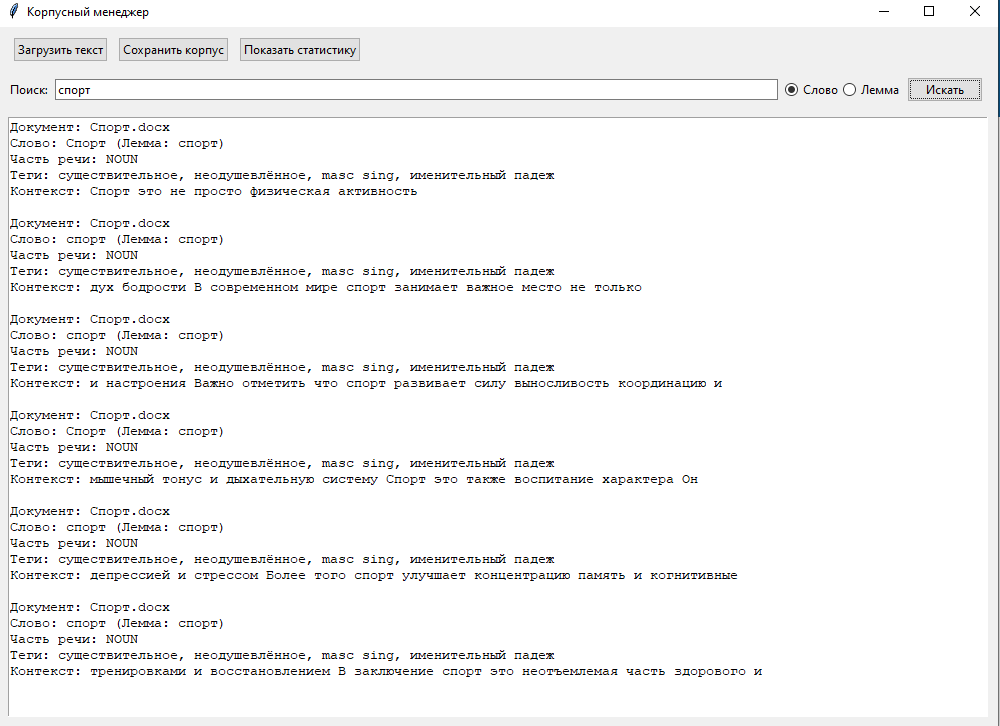


Рисунок 1: Интерфейс приложения

# Структура хранения данных

Данные хранятся в списке словарей (corpus), где каждый словарь представляет документ:

* • text: Исходный текст.
* • tokens: Список токенов с информацией (слово, лемма, часть речи, теги).
* • metadata: Метаданные (название, источник, дата, тип).

Индексы:

* • word\_index: Индекс слов для поиска по подстроке.
* • lemma\_index: Индекс лемм для поиска по нормализованным формам.

Пример структуры:

[

{

"text": "Спорт очень важен для здоровья",

"tokens": [

{

"word": "Спорт",

"lemma": "спорт",

"pos": "NOUN",

"tags": "существительное, неодушевлённое, masc sing, именительный падеж"

},

{

"word": "очень",

"lemma": "очень",

"pos": "ADVB",

"tags": "наречие"

},

{

"word": "важен",

"lemma": "важный",

"pos": "ADJS",

"tags": "прилагательное (краткое), Qual masc, единственное число"

},

{

"word": "для",

"lemma": "для",

"pos": "PREP",

"tags": "предлог"

},

{

"word": "здоровья",

"lemma": "здоровье",

"pos": "NOUN",

"tags": "существительное, неодушевлённое, neut sing, родительный падеж"

}

],

"metadata": {

"title": "1111.txt",

"source": "Загруженный файл",

"date": "2025-05-15",

"type": "unknown"

}

}

]

# Структурная схема приложения

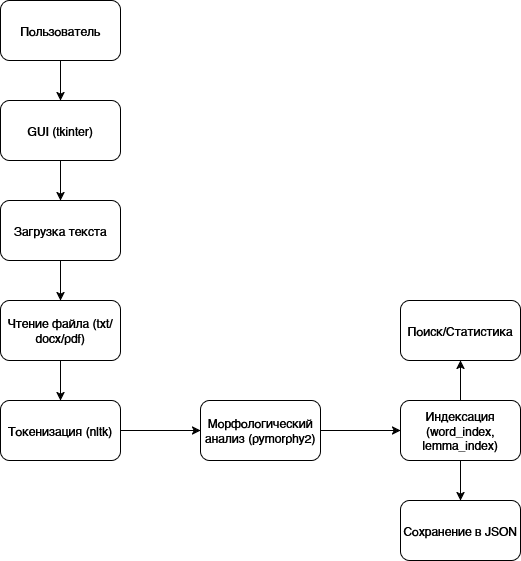


Рисунок 2: Структурная схема приложения

# Описание алгоритмов

## Добавление текста в корпус

1. Пользователь выбирает файл (TXT, DOCX, PDF).
2. Текст извлекается с помощью соответствующей библиотеки (txt, python-docx, pdfplumber).
3. Текст токенизируется (nltk).
4. Для каждого токена выполняется морфологический анализ (pymorphy2).
5. Токены индексируются (word\_index, lemma\_index).
6. Документ сохраняется в корпус с метаданными.

## Поиск по корпусу

1. Пользователь вводит запрос и выбирает тип поиска (слово/лемма).
2. Запрос ищется в соответствующем индексе.
3. Для каждого совпадения извлекается контекст и метаданные.
4. Результаты отображаются в GUI.

# Тест производительности

Тест проводился на корпусе из 10 документов (5000 слов). Среднее время выполнения на стандартном ПК:

* Добавление документа: 1,8 с.
* Поиск по слову: 0,03 с.
* Получение статистики: 0,05 с.

Основное время занимает морфологический анализ.

# Вывод

Разработан корпусный менеджер для текстов в области спорта. Приложение поддерживает загрузку текстов (TXT, DOCX, PDF), морфологический анализ, поиск по словам и леммам, а также статистику. Интерфейс интуитивно понятен. Перспективы: добавление поддержки других языков, расширение типов разметки, интеграция с базами данных.