ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук
Департамент программной инженерии

СОГЛАСОВАНО
Доцент департамента программной инженерии факультета компьютерных наук, канд. техн. наук.
_____ Х.М. Салех
«13» _____05_____ 2023 г.

УТВЕРЖДАЮ
Академический руководитель образовательной программы «Программная инженерия» профессор департамента программной инженерии, канд. техн. наук
_____ В.В. Шилов
«13» _____05_____ 2023 г.

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл	

УЛУЧШЕНИЕ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА РУССКОГО УЧЕБНОГО КОРПУСА

Текст программы

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.09.11-01 12 01-1-ЛУ

Исполнитель
студент группы БПИ 217
_____ / М.Д. Пересторонин/
«13»_____05_____ 2023 г.

Москва 2023

УТВЕРЖДЕН
RU.17701729.09.11-01 12 01-1-ЛУ

УЛУЧШЕНИЕ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА РУССКОГО УЧЕБНОГО КОРПУСА

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл	

Текст программы

RU.17701729.09.11-01 12 01-1

Листов 9

Москва 2023

АННОТАЦИЯ

В данном документе приведены ссылки на код, выполняющий следующие задачи:

1. Получения данных из базы данных
2. Хранение данных в базе данных (Django модели)
3. Токенизация и морфологический разбор текстов с применением машинного обучения
4. Сбор, обработка, отображение и визуализация статистических данных проекта
5. Анимирование статических элементов страницы

Среда разработки – PyCharm.

Настоящий документ разработан в соответствии с требованиями:

1) ГОСТ 19.101-77 Виды программ и программных документов;
2) ГОСТ 19.103-77 Обозначения программ и программных документов;
3) ГОСТ 19.104-78 Основные надписи;
4) ГОСТ 19.105-78 Общие требования к программным документам;
5) ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом
6) ГОСТ 19.505-79 ГОСТ 19.401-78 Текст программы. Требования к содержанию и оформлению. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

Изменения к данному Руководству оператора оформляются согласно ГОСТ 19.603-78 [8], ГОСТ 19.604-78

СОДЕРЖАНИЕ

1. ВВЕДЕНИЕ

1.1. Наименование программы

Наименование программы – «Русский Учебный Корпус» («Russian Learner Corpus»).

Краткое наименование программы – «RLC».

2. ТЕКСТ ПРОГРАММЫ

Репозиторий программы «Русский учебный корпус» находится по ссылке https://github.com/shar3nda/rlc_new. В корневой директории находятся файлы .editorconfig (конфигурация форматирования в проекте), .gitignore (файл игнорируемых файлов в системе контроля версий Git), Dockerfile и nginx.Dockerfile (файлы, описывающие контейнеры Docker), README.md (файл с информацией о проекте), docker-compose.yml (файл, описывающий связь и взаимодействие различных контейнеров Docker), manage.py (сгенерированный Django файл для взаимодействия с проектом), nginx.conf (файл конфигурации веб-сервера nginx), requirements.txt (файл со списком python-зависимостей проекта) и seed.py (скрипт для генерации тестовых данных). Также, в корневой директории находятся следующие поддиректории:

1. api (https://github.com/shar3nda/rlc_new/tree/master/api) – Django-приложение, содержащее API сайта для связи бекэнда и фронтенда
2. auto_annotator (https://github.com/shar3nda/rlc_new/tree/master/auto_annotator) – Python-библиотека для авторазметки текстов
3. content (https://github.com/shar3nda/rlc_new/tree/master/content) – Django-приложение, содержащее отображения и модели для новостных статей и главной страницы
4. corpus (https://github.com/shar3nda/rlc_new/tree/master/corpus) – Django-приложение, содержащее основные модели и логику приложения
5. data_import (https://github.com/shar3nda/rlc_new/tree/master/data_import) – Python-утилита для импорта документов из json-файла
6. locale (https://github.com/shar3nda/rlc_new/tree/master/locale) – папка с информацией о локали и переводах
7. rlc_new (https://github.com/shar3nda/rlc_new/tree/master/rlc_new) – настройки Django-проекта
8. static (https://github.com/shar3nda/rlc_new/tree/master/static) – статические файлы проекта (.js, .css и изображения)
9. templates (https://github.com/shar3nda/rlc_new/tree/master/templates) – html-шаблоны страниц сайта

2.1. Модели

2.1.1. content/models.py

https://github.com/shar3nda/rlc_new/blob/master/content/models.py

2.1.1.1. class Section - класс для хранения секций главной страницы. Содержит следующие поля:

2.1.1.1.1. text_rus – текст на русском языке
2.1.1.1.2. text_eng – текст на английском языке
2.1.1.1.3. header_rus – название на русском
2.1.1.1.4. header_eng – название на английском
2.1.1.1.5. number – номер секции на странице

2.1.2. corpus/models.py

https://github.com/shar3nda/rlc_new/blob/master/corpus/models.py

2.1.2.1. class Author – класс для хранения информации об авторе. Содержит следующие поля:

2.1.2.1.1. name – имя автора
2.1.2.1.2. gender – пол автора
2.1.2.1.3. program – программа автора
2.1.2.1.4. language_background – тип носителя (эритажный/иностранный)
2.1.2.1.5. dominant_language – доминантный язык автора
2.1.2.1.6. favorite – является ли автор сохраненным
2.1.2.1.7. Source – откуда взяты текста, написанные этим автором

2.1.2.2. class Document – класс, содержащий текст для аннотации, а также его метаданные. Содержит следующие поля

2.1.2.2.1. title – название документа
2.1.2.2.2. user – сотрудник, который добавил текст
2.1.2.2.3. created_on – дата, когда был добавлен документ в корпус
2.1.2.2.4. date – дата написания текста
2.1.2.2.5. genre – жанр текста
2.1.2.2.6. language_level – уровень владения языком автора
2.1.2.2.7. subcorpus – подкорпус
2.1.2.2.8. status – статус текста (не аннотирован, аннотирован, проверен)
2.1.2.2.9. author – автор текста
2.1.2.2.10. time limit – ограничение по времени
2.1.2.2.11. oral – является ли текст устным
2.1.2.2.12. annotators – аннотаторы
2.1.2.2.13. функция serialize для экспорта

2.1.2.3. class Sentence – класс для хранения информации о предложении в тексте. Содержит следующие поля

2.1.2.3.1. document – документ к которому относится предложение
2.1.2.3.2. text – текст предложения
2.1.2.3.3. markup – разметка для морфологического разбора
2.1.2.3.4. number – номер предложения в тексте

2.1.2.3.5. функция get_correction – возвращает откорректированную версию текста

2.1.2.3.6. функция serialize для экспорта

2.1.2.4. class Annotation – класс для хранения информации об аннотации. Содержит следующие поля:

2.1.2.4.1. document – документ к которому относится аннотация

2.1.2.4.2. sentence – предложение к которому относится аннотация

2.1.2.4.3. user – сотрудник, который добавил

2.1.2.4.4. guid – уникальный идентификатор аннотации в тексте

2.1.2.4.5. json – объект аннотации в формате json

2.1.2.4.6. alt – является ли исправление альтернативным

2.1.2.4.7. функция serialize для экспорта

2.1.2.5. class Token - класс для хранения информации о токене (слове). Содержит следующие поля:

2.1.2.5.1. token – сам токен (слово)

2.1.2.5.2. document – номер текста, к которому относится слово

2.1.2.5.3. sentence – номер предложения, к которому относится слово

2.1.2.5.4. start – начальная позиция слова в предложении

2.1.2.5.5. end – конечная позиция слова в предложении

2.1.2.5.6. number – номер слова в предложении

2.1.2.5.7. pos – часть речи

2.1.2.5.8. feats – грамматические характеристики

2.1.2.5.9. lemma – начальная форма слова

2.2. Токенизация

https://github.com/shar3nda/rlc_new/blob/master/auto_annotator/text_processor.py
class TextProcessor - проводит предобработку текста, делит его на токены (слова). К каждому токену применяется морфологический разбор и получение леммы.

2.3. Статистика

views.py - Получение данных и подсчет для статистики:
https://github.com/shar3nda/rlc_new/blob/master/corpus/views.py
statistics.html:
Отображение статистики и анимация ее элементов:
https://github.com/shar3nda/rlc_new/blob/master/templates/statistics.html
animateCounter – анимация подсчета документов/текстов/предложений

3. СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1) ГОСТ 19.101–77 Виды программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
2) ГОСТ 19.102–77 Стадии разработки. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
3) ГОСТ 19.103–77 Обозначения программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
4) ГОСТ 19.104–78 Основные надписи. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
5) ГОСТ 19.105–78 Общие требования к программным документам. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
6) ГОСТ 19.106–78 Требования к программным документам, выполненным печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
7) ГОСТ 19.603–78 Общие правила внесения изменений. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
8) ГОСТ 19.604–78 Правила внесения изменений в программные документы, выполненные печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
9) ГОСТ 19.401–78 Текст программы. Требования к содержанию и оформлению. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

Изм.	Номера листов (страниц)				Всего листов (страниц) в документе	№ документа	Входящий № сопроводительного документа и дата	Подпись	Дата
	измененных	замененных	новых	аннулированных					