Проектирование и разработка ПО

Для реализации программного модуля использовались следующие сторонние открытые библиотеки:

- Pandas (библиотека Python для обработки и анализа структурированных данных);
- Docker (инструмент контейнеризации приложений);
- Flask (фреймворк Python для создания веб-приложений);
- Re (модуль Python для поддержки регулярных выражений);
- Requests (библиотека, позволяющая отправлять запросы HTTP в Python);
- Kioskboard (библиотека JavaScript для использования виртуальных клавиатур);
- DevBridge Autocomplete (библиотека для создания полей автоматического заполнения/подсказок для полей ввода текста);
- Bootstrap Icons (библиотека с набором иконок).

ПО реализовано как единое веб-приложение, фронтенд реализован на технологическом стеке JavaScript/Python.

На данный момент агрегатор корпусов способен поддерживать работу с большим числом корпусов на платформе Tsakorpus, что позволяет уже делать некоторые типологические исследования, опираясь на данные корпусов малых языков.

Программное обеспечение, необходимое для функционирования программы

Для функционирования программы необходимы:

1. операционная система Ubuntu не ниже 16.04;
2. интерпретатор Python версии не ниже 3.7;
3. доступ к сети Интернет.

Языки программирования, на которых написана программа

Исходным языком программирования для программного модуля является Python версии 3.7. Языком верстки интерфейса является HTML.

Среда разработки – VS Code.

Алгоритм, лежащий в основе программы

1. подгрузка перечня корпусных ресурсов, работающих в системе TSAKorpus;
2. проверка доступности данных ресурсов;
3. получение уникальных токенов сессии для каждого работающего корпуса;
4. Извлечение информации о доступных поисковых модулях на ресурсах;
5. Определение общего подмножества поисковых модулей, пригодных для интегрированной работы;
6. Создание графического интерфейса для осуществления поиска;
7. Обработка пользовательского поискового запроса, сформулированного в интерфейсе, путём выполнения серии запросов к соответствующим корпусным менеджерам для получения данных, отвечающих первоначальному запросу пользователя;
8. Интеграция полученной информации из различных источников в единый интерфейс выдачи с поддержкой пагинации;
9. Формирование файлов для скачивания результатов в оригинальном формате платформы TSAKorpus;
10. Создание файлов JSON для скачивания полных результатов в машиночитаемом формате.

Требуемые директории и файлы в составе файловой системы

- Репозиторий framework:
    - Директория static: Содержит ресурсы, необходимые для работы пользовательского интерфейса.
        - Поддиректории:
            - autocomplete_devbridge: Файлы для автозаполнения.
            - bootstrap-5.0.0-beta1-dist: Файлы Bootstrap для стилизации.
            - css: Каскадные таблицы стилей.
            - d3: Библиотека D3.js для визуализации данных.
            - icons-1.4.1: Иконки для интерфейса.
            - img: Изображения.
            - jquery-3.5.1-dist: Библиотека jQuery.
            - js: Скрипты JavaScript.

            - keyboards: Виртуальные клавиатуры.
            - kioskboard-1.4.0: Файлы для реализации киоск-интерфейсов.
            - vendor: Дополнительные сторонние библиотеки и плагины.
    - Директория templates: Хранит HTML-шаблоны для различных частей веб-интерфейса.
        - Поддиректории:
            - admin: Шаблоны для административной панели.
            - `index`: Главная страница.
            - `modals`: Модальные окна.
            - `search_results`: Страница с результатами поиска.
        - Файлы:
            - footer.html: Шаблон подвала страницы.
            - fulltext.html: Шаблон для отображения полного текста.
            - head_add.html: Дополнительные элементы в `<head>`.
            - header.html: Шаблон заголовка страницы.
            - index.html: Шаблон главной страницы.
            - query_area.html: Шаблон области ввода запросов.
    - app.py: Основной файл приложения на Python, запускающий сервер и обрабатывающий запросы.
- Корневые файлы:
    - .gitmodules: Конфигурационный файл для Git submodules.
    - Dockerfile: Файл для контейнеризации приложения.
    - README.md: Документация по установке и использованию приложения.
    - docker-compose.yml: Файл для развертывания приложения с помощью Docker Compose.

Эти директории и файлы обеспечивают полноценное функционирование системы, предоставляя необходимые инструменты для контейнеризации, настройки и развертывания компонентов системы, а также для создания и поддержки пользовательского интерфейса.

Тестирование

ПО было протестировано на данных корпусов языков России в 2023 году. В частности, была протестирована работа со следующими корпусами языков: адыгейский, новый арамейский урмийский, новый арамейский туройо, чукотский.
[1] http://adyghe.web-corpora.net/adyghe_corpus/search
[2] http://neo-aramaic.web-corpora.net/urmi_corpus/search
[3] http://neo-aramaic.web-corpora.net/turoyo_corpus/search
[4] https://chuklang.ru/corpus

Приобретение и поставка ПО

Разработка была ПО осуществлена на базе Центра искусственного интеллекта НИУ ВШЭ, который регулирует приобретение полного пакета ПО или его отдельных модулей. ПО запланировано к размещению в открытом свободном доступе.

Документирование

Документирование ПО осуществлено его разработчиками. Прилагаемая к дистрибутиву ПО документация описывает назначение, структуру, установку, эксплуатацию и техническую поддержку

Обучение и квалификация персонала

Для эксплуатации ПО требуется наличие администратора, в функции которого входит модерация новых источников данных, которые могут быть добавлены в ПО, а также мониторинг инфраструктуры, необходимой для функционирования ПО. Дополнительных специальных условий эксплуатации, кроме соблюдения минимальных требований к вычислительным ресурсам технических средств, на которых ПО развернуто, не требуется. Пользователь ПО должен быть уверенным пользователем ПК.

Особого обслуживания ПО не требуется.

Разработка ПО и его техническая поддержка осуществляется сотрудниками Центра искусственного интеллекта НИУ ВШЭ и Международной лаборатории биоинформатики Факультета компьютерных наук НИУ ВШЭ, обладающими необходимой для этого квалификацией. Вся инфраструктура разработки, а также разработчики ПО,

осуществляющие техническую его поддержку, размещаются по адресу: АУК "Покровский бульвар", Москва, Покровский б-р, д. 11

Поддержка версий и доработка

Доработка ПО в случае добавления новых источников для поиска, предполагается соответствующим редактированием соответствующей части параметров основного модуля и входит в функции администратора со стороны пользователя.

Устранение сбойных ситуаций

Устранение сбойных ситуаций при развертывании ПО на серверном оборудовании, соответствующем перечисленным в документации требованиям, осуществляется силами технической поддержки.

При развертывании ПО на серверном оборудовании должно быть предусмотрено бесперебойное питание технических средств.

В случае использования различных серверов (таких как сервер приложений, сервер для хранения объектов на файловой системе) должны быть проведены мероприятия по обеспечению и поддержке сетевой связности между серверами. Должен быть обеспечен доступ в сеть интернет.

Время восстановления после отказа, вызванного сбоемэлектроснабжения, сетевого оборудования, программных средств, нефатальным сбоем централизованного хранилища данных, не должно превышать трех суток.

Время восстановления после отказа, вызванного фатальным сбоем централизованного хранилища данных, который привел к логическому или физическому разрушению указанного хранилища, не должно превышать времени, необходимого на восстановление централизованного хранилища данных из резервной копии.

Время восстановления после отказа, вызванного неисправностью технических средств, не должно превышать времени, необходимого на устранение неисправностей или замену технических средств.

На этапах разработки программы разработчиком предусмотрена невозможность возникновения отказов в работе программы из-за некорректных действий оператора.