Учреждение образования
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Кафедра интеллектуальных информационных технологий

Отчет по лабораторной работе №1 по курсу «Естественно-языковой интерфейс интеллектуальных систем»

Выполнили студенты группы 121701: Мулярчик Д. С.

Леманович Д. К

Проверил: Крапивин Ю.Б.

Минск 2024

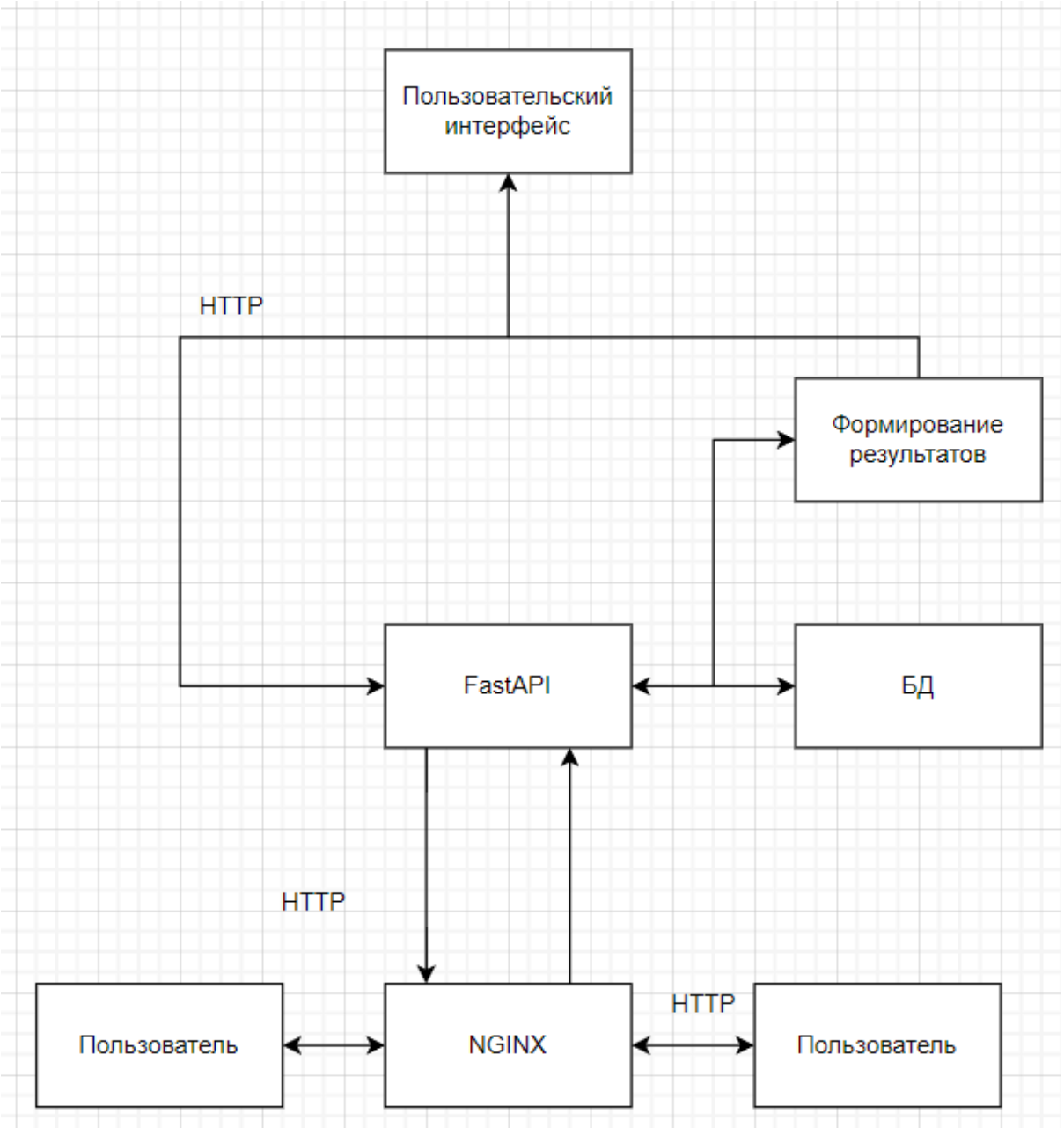
Цель работы

Освоить на практике основные принципы реализации информационно-поисковых систем и методы оценки качества их работы.

Требования к разрабатываемой системе

- на входе – множество естественно-языковых текстов по которым осуществляется поиск;
- система должна позволять пользователю формулировать ЕЯ-запрос;
- на выходе – список документов, релевантных запросу пользователя,в соответствие с моделью поиска, согласно варианту;
- результаты поиска должны содержать: активную ссылку на документ, список слов запроса присутствующих в документе.
- на основании информации о существующих метриках, наиболее часто использующихся для оценки качества работы систем информационного поиска (см. 2004_romip_metrix.pdf), требуется дать оценку работы СИП. Вычисление оценок, получаемых на основании метрик, реализовать программно, путем вызова соответствующего подменю, и отображать в виде таблиц и графиков;
- интерфейс системы должен быть предельно простым и доступным для пользователей любого уровня, содержать понятный набор инструментов и средств, а также help-средства.

Структура системы



1. На каждом устройстве в локальной сети работает процесс, который в зависимости от путей определенных в конфигурации системы для отслеживания, определяет изменения в файлах и отправляет запросы на сервер для изменения индексов.
2. Система получает новое содержимое файла.
3. Если такого файла еще нет - разбиваем его на сниппеты и каждый сниппет индексируем и помещаем в базу данных, если такой файл уже существует - удаляем старые индексы и создаем новые.

4. При запросе пользователя проверяем индексы по которым содержится нужный нам сниппет.
5. Собираем все нахождения в список, где отображаем активную ссылку до файла и найденный сниппет.

Основные алгоритмы реализации компонентов

Алгоритм индексации документов

1. Создается схема Schema, которая определяет структуру индекса. В данном случае она включает три поля:
    a. title: текстовое поле, где хранится заголовок документа (хранится в индексе).
    b. path: уникальный идентификатор документа (хранится в индексе).
    c. content: текстовое поле для содержания документа (также хранится в индексе).
2. Проверяется, существует ли директория indexdir, где будет храниться индекс. Если директория не существует, она создается.
3. С помощью функции create_in создается индекс в директории indexdir, используя ранее определенную схему.
4. Открывается контекстный менеджер (with) для создания писателя (writer) индекса, который позволяет добавлять документы в индекс. В цикле перебираются все документы в переданном словаре documents, где ключом является путь к файлу (например, имя документа), а значением — его содержани. Для каждого документа:
    a. Заголовок (title) извлекается из имени файла, убирая расширение (например, "document.txt" становится "document").
    b. Документ добавляется в индекс с помощью метода add_document в который передаются заголовок, путь и содержание документа.
5. Функция возвращает объект индекса (ix), который теперь содержит все добавленные документы и готов к поисковым операциям.

Алгоритм работы поиска в индексе

1. Создается пустой список results_list, в который будут добавляться результаты поиска.
2. В зависимости от значения logic_operator, выбирается группа логики поиска:
    a. Если logic_operator равен "AND", используется AndGroup, что означает, что результаты должны соответствовать всем условиям запроса.
    b. Если "OR", используется OrGroup, что означает, что результаты могут соответствовать любому из условий.
3. Открывается контекстный менеджер для поиска (with ix.searcher() as searcher), который создает объект searcher для выполнения поисковых операций.
   Создается объект MultifieldParser, который позволяет разбирать запрос query_str по нескольким полям (в данном случае по полям "title" и "content") с использованием заданной схемы (schema=ix.schema) и выбранной группы логики.
4. Запрос query_str передается парсеру для преобразования в объект запроса (query = parser.parse(query_str)).
5. Метод searcher.search(query) выполняет поиск по индексу с использованием созданного запроса. Результаты сохраняются в переменной results.
6. В цикле перебираются все найденные результаты в results
   Для каждого результата извлекается выделенное содержание из поля content с помощью метода result.highlights('content').
    a. title: заголовок документа.
    b. path: путь к документу.
    c. highlighted: выделенное содержание.
7. Функция возвращает список results_list, содержащий все найденные результаты, включая заголовки, пути и выделенное содержание.

Результаты оценки по метрикам

Примем во внимание следующие условные обозначения:

	релевантны	не релевантны
найдено системой	a	b
не найдено системой	c	d

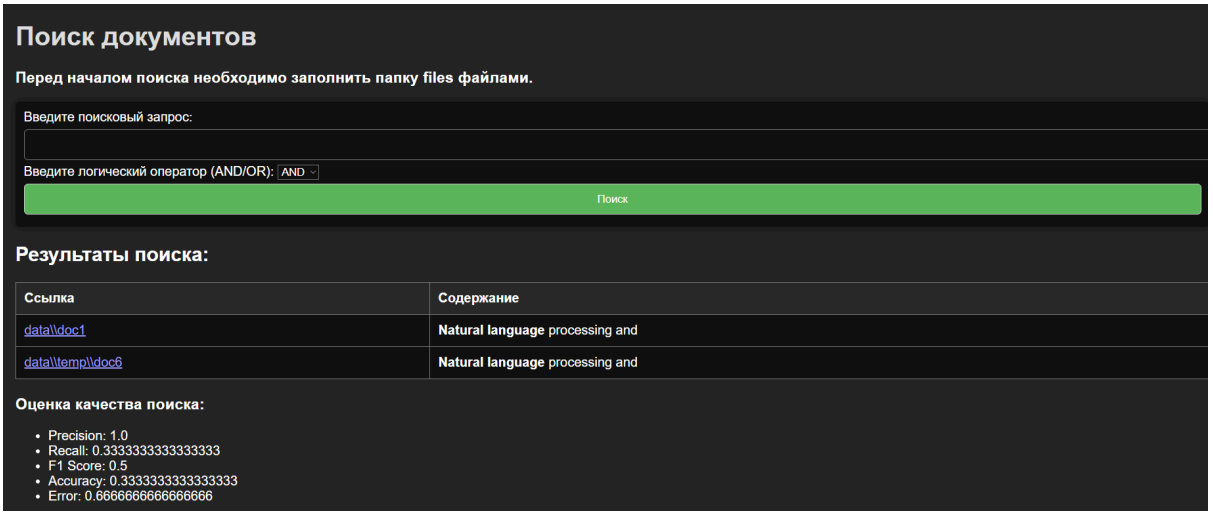
Полнота - вычисляется как отношение найденных релевантных документов к общему количеству релевантных документов. Характеризует способность алгоритма обнаруживать релевантные документы в принципе.

Точность – вычисляется как отношение найденных релевантных документов к общему количеству найденных документов. Характеризует способность алгоритма отличать релевантные документы от нерелевантных.

Аккуратность – вычисляется как отношение правильно принятых системой решений к общему числу решений. Характеризует способность алгоритма делать правильные заключения относительно релевантности/нерелевантности документа.

Ошибка – вычисляется как отношение неправильно принятых системой решений к общему числу решений. Характеризует неспособность алгоритма делать правильные заключения относительно релевантности/нерелевантности документа.

F-мера часто используется как единая метрика, объединяющая метрики полноты и точности в одну метрику



Анализ и предложения по улучшению работы СИП

Эффективность поиска можно оценить опираясь на качественные и количественные характеристики найденных документов. Качественные характеристики измеряются уровнем соответствия выданных релевантных документов запросу, а количественные - соотношением выданных и не выданных релевантных документов.

Значение полноты составляет 0.33, что означает, что алгоритм находит лишь одну треть релевантных документов. Это указывает на то, что две трети релевантных документов остаются незамеченными. F1 Score равен 0.5, что является средней оценкой между точностью и полнотой. F1 Score показывает компромисс между точностью и полнотой, и низкий уровень этого показателя также указывает на возможность для улучшений.

Предложения по улучшению работы СИП:

- Использование кэша для ускорения поиска документов, которые часто появляются в поиске.

Описание готовых к использованию компонентов

- Веб-фреймворк FastAPI использовался для представления интерфейса к системе, а также для общения в локальной сети всех устройств.

- Whoosh библиотека для Python, предназначенная для индексирования и поиска текста.
- Sclearn библиотека для подсчета метрик.