Общий принцип взаимодействия между службами на примере стандартного сценария обработки запроса от пользователя

1. Получение запроса от пользователя:

Пользователь отправляет запрос через интерфейс общения (например, Telegram), и затем служба обработки запросов получает этот запрос и выполняет анализ.

2. Анализ запроса.

Служба обработки запросов анализирует текст вопроса пользователя, используя модели машинного обучения, и определяет категорию этого вопроса. Например, если вопрос относится к категории обучения, служба определяет это и перенаправляет вопрос дальше к службе управления данными.

ВАЖНО:

- Если вопрос относится к двум и более категориям, необходимо задать уточняющий вопрос. (Как это реализовать, пока не ясно)

3. Обработка запроса в службе базы знаний.

Служба управления данными получает запрос и извлекает соответствующий ответ из базы данных, используя ORM и алгоритм поиска. (?)

**ВОЗМОЖНОЕ РЕШЕНИЕ!!!!!!**

**Хранить токены TF-IDF для ответов в базе данных и искать 10-15 ответов согласно схожести токенов с вопросом (из вопроса извлекается TF-IDF ключевые слова для сравнивания с TF-IDF ключевыми словами ответов в базе данных) и фильтруя по категории вопроса.**

~~ПРОБЛЕМЫ СУД:~~

~~- Неэффективно искать по всей базе данных даже через алгоритм TF-IDF. Какой есть способ быстрого и удобного поиска по БД?~~

~~ВОЗМОЖНЫЕ РЕШЕНИЯ:~~

~~- Хранить ключевые слова для каждого вопроса (но как их получить в автоматическом режиме и как в дальнейшем выполнять по ним поиск?)~~

~~- При поиске в базе данных отсекать все по категории вопроса~~

~~- Индексировать ключевые слова (а может не слова, а вектора слов???)~~

~~ВОПРОСЫ К П.3:~~

~~- А как ищет гугл или яндекс?~~

4. Отправка ответа пользователю.

Служба обработки запросов получает ответ от службы управления данными и отправляет его пользователю через интерфейс общения (например, ответ в Telegram).

5. В процессе взаимодействия служба диалога отслеживает контекст общения с пользователем, сохраняя информацию о предыдущих запросах и ответах. В случае необходимости служба диалога может задавать уточняющие вопросы пользователю и обеспечивать переход к живому куратору.

ВОПРОСЫ К П.5:

- Как определить, когда необходимо переходить к живому куратору?

Ответ: если показатель релевантности ниже определенного уровня, мы отдаем ответ и вместе с этим , а затем уточняем у пользователя, дан ли ответ. Если ответ дан положительный – радуемся🥰💅🎉 Если же нет, то спрашиваем желает ли пользователь уточнить вопрос или связаться с живым куратором.

РЕСУРСЫ:

* Извлечение признаков из текстовых данных с использованием TF-IDF: <https://habr.com/ru/companies/otus/articles/755772/>
* Алгоритмы для выделения ключевых слов: Rake, YAKE!, TextRank: <https://newtechaudit.ru/algoritmy-dlya-vydeleniya-klyuchevyh-slov-rake-yake-textrank/>
* Расширяем существующий scikit-learn классификатор: <https://makesomecode.me/2020/04/extend-sklearn-classifier/>
* Применение нейросетевых методов извлечения ключевых слов для составления резюме студента по рабочим программам: <https://scm.etu.ru/assets/files/2023/sbornik/186-189.pdf>
* ICF, the missing ingredient of TF-IDF: <https://medium.com/codex/icf-the-missing-ingredient-of-tf-idf-d9f715c9946f>
* Реализация поискового движка с ранжированием на Python (Часть 3): <https://habr.com/ru/articles/263983/>
* Семантический анализ для автоматической обработки естественного языка: <https://rdc.grfc.ru/2021/09/semantic_analysis/>

Идеи по реализации

БАЗА ДАННЫХ

Выбор для хранения базы знаний стоял между двумя видами баз данных – не реляционной колоночной (данные хранятся в виде разреженной матрицы, где строки и столбцы выступают в качестве ключей) и реляционной (стандартная).

1) ClickHouse — столбцовая система управления базами данных (СУБД) для онлайн-обработки аналитических запросов (OLAP).

Основные особенности OLAP-сценариев интересные для нас:

- подавляющее большинство запросов — на чтение;

- данные обновляются достаточно большими пачками (> 1000 строк), а не по одной строке, или не обновляются вообще;

- данные добавляются в БД, но не изменяются;

- транзакции отсутствуют.

Минусы:

- сложно убедиться, что данные не дублируются, но есть определенные способы для дедупликации (параметр в настройках таблицы)

2) PostgresSQL – строковая система управления базами данных (СУБД) для онлайн обработки транзакций (OLTP).

Основные особенности OLTP-сценариев интересные для нас:

- данные часто создаются, изменяются и удаляются;

- присутствуют транзакции;

- подавляющее большинство запросов — на запись.

Мы выбрали Clickhouse для хранения нашей базы знаний, так как операции чтения являются более приоритетным для нас, чем наличие транзакций и частоты записи. В ходе работы бота – мы чаще читаем большие объемы данных.

Ресурсы:

- Артем Шутак — Вставить в ClickHouse и не умереть: <https://www.youtube.com/watch?v=XKBYYP5k_Uo>

- Что такое ClickHouse? <https://clickhouse.com/docs/ru>