**2. Векторные базы данных.**

После завершения деструктуризации данных по Книгам и Приложениям СКИОВО для бассейнов рек (Дон, Терек, Сура, Волга, Кубань, Обь и Урал), необходимо определить место для хранения текстовой части, извлеченной из этих документов. В ходе деструктуризации для каждого документа был создан JSON-файл, который содержит четкую структуру разделов и подразделов, а также включает метаданные, такие как номера начальных и конечных страниц, названия глав и уровень глубины разбивки.

Для выбора подходящего места хранения данных важно учитывать, что главная цель проекта — создание удобной вопросно-ответной системы по СКИОВО. Это предполагает возможность быстрого и точного поиска информации на основе содержимого книг и приложений, а также поддержки семантического поиска по контексту. В связи с этим было предложено использовать векторную базу данных, основанную на деструктурированных данных.

**Использование векторов в машинном обучении**

Векторы играют ключевую роль в машинном обучении, так как именно с ними компьютеры могут работать и воспринимать данные. Любые изображения или текст в начале преобразуются в числовой вид, что позволяет алгоритмам машинного обучения обрабатывать эти данные. Этот процесс векторизации часто называется созданием **эмбеддингов** (от англ. embedding) — векторами в виде массивов чисел, которые генерируются с помощью языковых моделей.

**Векторизация слов**

Слова и другие элементы данных располагаются в пространстве в виде векторов, что позволяет моделям изучать их отношения и семантические связи. Чем ближе два вектора друг к другу, тем более схожими по значению считаются соответствующие слова. Например, слова «король» и «королева» будут располагаться ближе друг к другу в многомерном пространстве, чем слова «король» и «яблоко».

Таким образом, векторы не только представляют слова, но и сохраняют информацию о их контексте и значении. Это позволяет языковым моделям выполнять множество задач, таких как:

* **Классификация текста**: Векторы помогают определять, к какой категории принадлежит текст (например, положительный или отрицательный отзыв).
* **Поиск по смыслу**: Модели могут находить слова или предложения, которые семантически близки к заданному запросу, даже если они не содержат одинаковых слов.
* **Синтаксический анализ**: Векторы могут использоваться для определения грамматических и синтаксических структур в тексте.

**Преимущества векторной базы данных**

При выборе векторной базы данных необходимо учитывать следующие аспекты:

1. **Индексация и фильтрация метаданных**: Векторные базы данных обеспечивают возможность быстрой индексации и эффективной фильтрации метаданных. Это позволяет извлекать нужные данные с высокой скоростью, что критично для вопросно-ответных систем, где важна скорость и точность обработки запросов.
2. **Локальное хранение**: Важно, чтобы выбранная база данных могла работать на локальных серверах, что обеспечит безопасность данных и контроль над инфраструктурой.
3. **Поддержка семантического поиска**: Возможность выполнять семантический поиск по векторным представлениям данных позволяет находить не только точные совпадения, но и близкие по смыслу ответы на запросы пользователей. Это значительно улучшает пользовательский опыт и эффективность системы.

**Выбор векторных баз данных**

Для реализации вышеописанных требований можно рассмотреть следующие решения:

* **Chroma**: Это высокопроизводительная векторная база данных, которая поддерживает эффективную индексацию и хранение векторов. Chroma позволяет выполнять семантический поиск и интегрируется с различными фреймворками машинного обучения, обеспечивая удобный интерфейс для работы с данными.
* **Qdrant**: Эта база данных оптимизирована для работы с векторами и предлагает мощные инструменты для индексации и поиска. Qdrant поддерживает фильтрацию по метаданным и может работать на локальных серверах, что делает ее хорошим выбором для проектов, требующих гибкости и контроля над данными.
* **FAISS (Facebook AI Similarity Search)**: FAISS — это библиотека для эффективного поиска и кластеризации больших коллекций векторов. Она предоставляет алгоритмы для быстрого поиска ближайших соседей и поддерживает индексацию, что позволяет выполнять семантический поиск по большим объемам данных. FAISS может быть интегрирован с другими системами, что делает его универсальным инструментом для обработки векторов.

**Процесс создания эмбедингов по json-файлам**

Для эффективного преобразования текста в векторы необходимо минимизировать потерю смысла. Прежде всего, требуется предварительная очистка данных — удаление лишних символов, таких как переносы строк, табуляции и другая служебная информация, которая не несет смысловой нагрузки.

После очистки возникает вопрос: какие объемы текста можно преобразовать в один вектор? В разных разделах может содержаться разное количество информации, и попытка преобразования, например, трёх страниц текста в один вектор размерности 300, скорее всего, приведет к потере смысла. В таком случае информация будет упрощена и обобщена, что может исказить исходное содержание данных.

Чтобы сохранить смысл при векторизации, необходимо не только очистить текст, но и разбить его на логически обоснованные фрагменты. Размер этих фрагментов должен быть оптимальным для того, чтобы модель могла эффективно извлечь информацию без потери ключевых деталей. Важным шагом является также выбор подходящей модели векторизации, которая наиболее эффективно работает с данными СКИОВО, учитывая специфику текстов и цели проекта.

Таким образом, качественное преобразование текста в векторы требует тщательной подготовки, включающей очистку данных, корректную разбивку текста и выбор оптимальной модели векторизации, которая бы соответствовала контексту и структуре документов. Если первые два шага относительно понятны и имеют чёткую методику выполнения, то процесс выбора модели векторизации является более сложным и специфичным. Это связано с необходимостью не только найти модель с подходящими характеристиками, но и протестировать её на данных СКИОВО, чтобы убедиться в её эффективности.

В ходе исследования рассматривались только открытые модели с платформы HuggingFace, представляющей собой обширное хранилище открытых моделей, предназначенных для решения различных задач.