Ниже описано, что содержится в каждом из этих файлов и для чего они используются:

1. **/content/drive/MyDrive/VideoRAG/features/example\_features.pt**  
   Этот файл содержит тензор эмбеддингов (векторов признаков), полученных из каждого извлечённого кадра видео.
   * **Как формируется:**  
     После извлечения кадров из видео, каждый кадр обрабатывается моделью CLIP (или другой выбранной моделью). Результатом работы модели является вектор признаков фиксированной размерности (например, 512 или 768 элементов).
   * **Структура:**  
     Это PyTorch-тензор размерности [N, D], где N — количество обработанных кадров, а D — размерность эмбеддинга.
   * **Назначение:**  
     Эти эмбеддинги используются для последующего поиска – по ним будет производиться сравнение с вектором запроса для определения наиболее похожих кадров.
2. **/content/drive/MyDrive/VideoRAG/features/example\_timestamps.pt**  
   Этот файл содержит информацию о временных метках для каждого извлечённого кадра.
   * **Как формируется:**  
     При извлечении кадров для каждого кадра сохраняется его имя (например, frame\_000123.jpg) и соответствующая временная метка (формат HH:MM:SS.mmm), которая вычисляется на основе FPS исходного видео.
   * **Структура:**  
     Обычно это объект, сохранённый через torch.save (например, DataFrame или другой список/словарь), где каждому кадру сопоставлена его временная метка.
   * **Назначение:**  
     При поиске по запросу результат (набор индексов наиболее похожих кадров) сопоставляется с временными метками. Это позволяет определить, в какой момент видео появляется искомая сцена или объект.
3. **/content/drive/MyDrive/VideoRAG/index/example.faiss**  
   Этот файл представляет собой построенный FAISS-индекс эмбеддингов.
   * **Как формируется:**  
     При помощи библиотеки FAISS из эмбеддингов, сохранённых в example\_features.pt, создаётся индекс (в данном случае используется IndexFlatIP – индекс для внутреннего произведения, часто применяемый для косинусного сходства после нормализации). Перед добавлением эмбеддингов к индексу их нормализуют (faiss.normalize\_L2).
   * **Структура:**  
     Это сериализованный индекс FAISS, который хранит все векторы признаков и позволяет быстро находить ближайших соседей (то есть кадры, наиболее похожие на заданный запрос).
   * **Назначение:**  
     При поиске по изображению (запросу) вектор запроса сравнивается с векторами из этого индекса, чтобы быстро найти топ-N наиболее похожих кадров. Полученные индексы затем используются для извлечения временных меток и визуализации результатов.

Таким образом, весь процесс выглядит следующим образом:

* **Извлечение кадров** → для каждого кадра создаётся эмбеддинг (сохраняется в example\_features.pt) и запоминается его временная метка (сохраняется в example\_timestamps.pt).
* **Построение индекса** → эмбеддинги заносятся в FAISS-индекс (example.faiss), который позволяет быстро искать похожие кадры.
* **Поиск** → по запросу (например, изображение) вычисляется эмбеддинг, проводится поиск по FAISS-индексу, и найденные результаты сопоставляются с временными метками для дальнейшей визуализации и вырезания сегментов видео.

Эта структура позволяет организовать быстрый поиск и навигацию по видео на основе семантических признаков, извлечённых из его кадров.