1. **Microservices for Data Preprocessing:**

   This microservice takes a CSV file as input, which contains raw data that needs to be preprocessed.

   It performs various preprocessing tasks on the data, such as cleaning, normalization, removing duplicates, and transforming the data into a suitable format for further processing.

   The preprocessed data can be stored in a structured format, such as a database, for easy retrieval and usage by other microservices.

2. **Microservices for Domain Graph:**

   This microservice takes CSV files as input and converts them into a knowledge graph.

   It analyzes the data in the CSV files and creates a graph representation where nodes represent entities (e.g., courses, topics, instructors) and edges represent relationships between these entities.

   The domain graph captures the semantic relationships among different entities, enabling better understanding and organization of the data.

3. **Microservices for SERP API:**

   This microservice interacts with a Search Engine Results Page (SERP) API, such as serpapi.com.

   It retrieves the links or URLs from the knowledge graph that represent relevant entities (e.g., courses, topics) and uses the SERP API to extract the actual SERP pages for those URLs.

   The SERP pages contain search results, metadata, and other information related to the URLs.

   The extracted SERP data is then saved in a NoSQL document database for further processing and retrieval.

4. **Microservices for Ranking:**

This microservice updates the ranks of URLs in the Search Engine Results Pages (SERP) based on user interactions.

It analyzes user behavior, such as clicks and engagement, to determine the relevance and popularity of URLs within the search results.

Using this information, the microservice updates the ranks of URLs, ensuring that the most relevant and useful content appears higher in the search results for better user experience.

5. **Keyword Prediction Microservice:**

   The keyword prediction microservice takes user search queries as input and predicts additional keywords that are relevant to the query.

   It can leverage machine learning models or natural language processing techniques to generate predictions based on patterns and associations in the input query.

   The predicted keywords can enhance the search functionality by expanding the user's query and improving the accuracy of search results.

6. **Scraper Microservice:**

   The scraper microservice is responsible for fetching and extracting data from various educational websites or data sources.

   It utilizes web scraping techniques or APIs provided by the target websites to fetch HTML content and extract relevant information, such as course details, descriptions, and instructor information.

   The extracted data is then processed, preprocessed, and stored in a structured format, ready to be indexed and searched by the edutech search engine.

Working Together of All Microservices

The data preprocessing microservice prepares the raw data for further processing, ensuring data quality and consistency.

The domain graph microservice organizes and represents the data in a semantic graph, capturing relationships among entities.

The SERP API microservice retrieves SERP pages for relevant URLs, enriching the search results with additional metadata.

The ranking microservice updates URL ranks based on user interactions, ensuring more relevant and popular results are displayed.

The keyword prediction microservice expands user queries, enhancing the search experience.

The scraper microservice fetches and extracts data from educational websites, providing fresh and relevant content for the search engine.

These microservices work together to preprocess, organize, fetch, and process data, improving the search experience for users.

Here's a step-by-step explanation of the order in which the microservices can work within your edutech search engine architecture:

1. **Microservices for Data Preprocessing:**

   This microservice works first as it preprocesses the raw data (CSV files) by cleaning, normalizing, and transforming it into a suitable format for further processing.

2. **Microservices for Domain Graph:**

   Once the data is preprocessed, the microservice for domain graph creation can work.

   It takes the preprocessed data and converts it into a knowledge graph, capturing relationships among entities.

3. **Scraper Microservice:**

   The scraper microservice can work after the domain graph microservice.

   It fetches and extracts data from various educational websites or data sources based on the defined scraping logic.

   The extracted data is then processed and stored, ready for integration into the search engine.

4. **Microservices for SERP API:**

   Once the data is available from the scraper microservice, the microservice for the SERP API can be employed.

   It takes the URLs from the knowledge graph, retrieves the corresponding SERP pages using the SERP API, and saves the extracted SERP data in a NoSQL document database.

5. **Microservices for Ranking:**

   The microservice for ranking can work in parallel with the SERP API microservice.

   It analyzes user interactions, such as clicks and engagement, to update the ranks of URLs within the search results based on their relevance and popularity.

6. **Keyword Prediction Microservice:**

   The keyword prediction microservice can work concurrently with the other microservices.

   It takes user search queries as input and predicts additional keywords relevant to the query to enhance the search functionality.

By following this order, the data preprocessing microservice lays the foundation for subsequent microservices. The domain graph microservice organizes the data, the scraper microservice fetches fresh content, the SERP API microservice retrieves search result data, the ranking microservice updates the ranks based on user interactions, and the keyword prediction microservice expands the search query. Together, these microservices collectively contribute to improving the search experience within your edutech search engine.