University Course Catalog Data Extraction and Query Challenge

Unleash the Course Catalog Crusher!

Attention all data detectives and code crusaders! Are you ready to take on the challenge of unlocking the secrets hidden within university course catalogs? These unstructured PDF documents hold a wealth of valuable information, but their true potential remains untapped. That's where you come in! 💡

In this exciting two-part challenge, you'll first develop a solution to extract key data elements from course catalog PDFs and transform them into structured, machine-readable formats. Then, you'll build a smart query engine that can provide precise answers to complex questions about courses, degree requirements, and academic policies. Your innovative solutions will have a real-world impact, streamlining processes and making life easier for students, faculty, and administrators alike. So, are you up for the challenge? Join us in cracking the course catalog code and unleashing the power of structured data! 💪📊

Objective

University course catalogs contain a wealth of information that is valuable to students, faculty, and administrators. However, this data is often locked in large, unstructured PDF documents that make it difficult to access and utilize effectively.

The goal of this two-part challenge is to first extract the key data elements from one or more university course catalog PDFs and convert them into a structured, machine-readable format.

Secondly, build a natural-language query interface that allows users to ask precise questions about courses, degree requirements, academic policies, etc. and receive accurate answers citing the specific PDF page and paragraph.

Unlocking this course catalog data in a structured, searchable format will enable new applications and automation of manual processes across the university.

Part 1: Course Catalog Data Extraction

Participants will be provided with one or more course catalog PDFs from various universities. The objective is to develop a solution to ingest a portion of these PDFs and extract key data elements as available within that portion. Possible examples:

- Course details: title, code, description, credit hours
- Degree requirements: required courses, electives, prerequisites
- Academic policies: grading system, academic standing, transfer credits

The extracted data should be output in standardized structured formats such as kv pairs, YAML/JSON, markdown, etc. as suitable, and stored into RDBMS and vector stores as appropriate. Solutions will be evaluated on the accuracy and completeness of the extracted data across the variations in catalog formats. Solutions should be generalizable to new, unseen university catalog formats.

Judging Criteria

1. Accuracy (40%)
    - Are the extracted data elements correct?
    - Is the extracted information complete?
    - Are data types (e.g., numbers, dates) correctly parsed?
2. Handling of PDF Elements (30%)
    - Does it successfully extract data from different elements (tables, enums)?
    - How robust is the solution to various elements?
3. Output Format (20%)
    - Is the output in a valid, structured format (kv, YAML, markdown)?
    - Is the output schema consistent and well-organized?
    - Are appropriate data types used?
4. System Design and Efficiency (10%)
    - How fast does the solution process the PDFs?
    - Is the code modular, well organized, well documented and easy to read?

Part 2: Course Catalog Query Engine

Using the structured course catalog data from Part 1, participants will build a natural language query interface that allows users to ask detailed questions about the catalog content and receive precise answers. Example queries:

- "What are the prerequisites for ECON 101 Intro to Microeconomics?"
- "How many credit hours are required for a BS in Computer Science? List the specific courses mentioned on page 58."

The query engine should return the specific answer to the question as well as the relevant PDF page number and paragraph citation. Submissions will be judged on the accuracy of the returned results, the ability to handle a variety of complex queries, and the user experience of the interface. Bonus points for handling contextual follow-up queries.

Judging Criteria

1. Query Result Accuracy (40%)
    - Does the system return the correct answer to the query?
    - Are the cited PDF page and paragraph references accurate?
2. Query Complexity Handling (30%)
    - Can the system handle a variety of query types (factoid, list, etc.)?
    - Does it successfully interpret complex query semantics?
    - Can it handle contextual follow-up queries?
3. User Interface and Experience (20%)
    - Is the query interface intuitive and easy to use?
    - Are the query results presented in a clear, understandable way?
    - Is the interface responsive and visually appealing?
4. System Design and Efficiency (10%)
    - Is the system responsive and low-latency?
    - Is the code modular, well organized, well documented and easy to read?

Assumptions

- All PDFs will be in English
- Catalog PDFs will be from U.S. universities
- No handwritten content, only text
- The input file format is PDF (no images, Excel, Word, etc.)

Source Document - VirginiaTech Catalog (Pages 1-100)

https://www.undergradcatalog.registrar.vt.edu/2018-19%20Undergraduate%20Catalog.pdf