CS 1671/2071
Human Language Technologies

Session 24: Information retrieval, RAG

Michael Miller Yoder

April 14, 2025

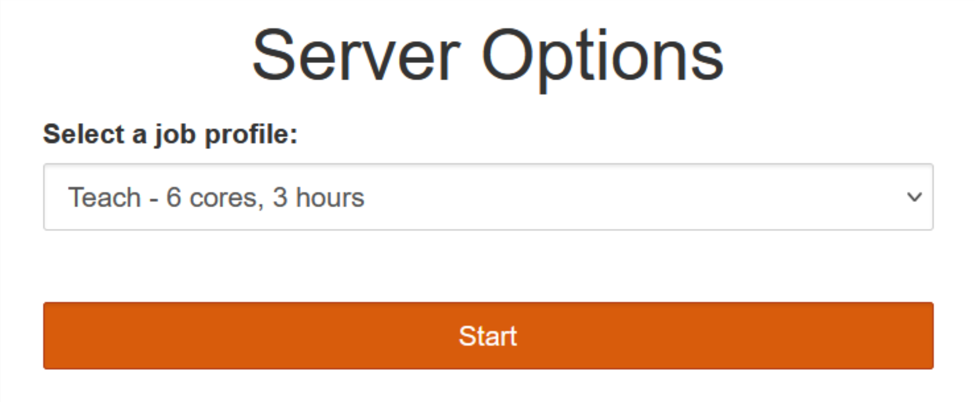
University of Pittsburgh | School of Computing and Information

Course logistics

- Homework 3 is **due today, Mon Apr 14 at 11:59pm**
  - See latest version of hw3_template.ipynb for updated parse_answer function that can handle negative numbers
- Final report due date extended to **Mon Apr 28**
  - Instructions will be released
- Presentations will be given during the final class session, **Apr 30, 12-1:50pm**

Prep and load packages for today's coding notebook

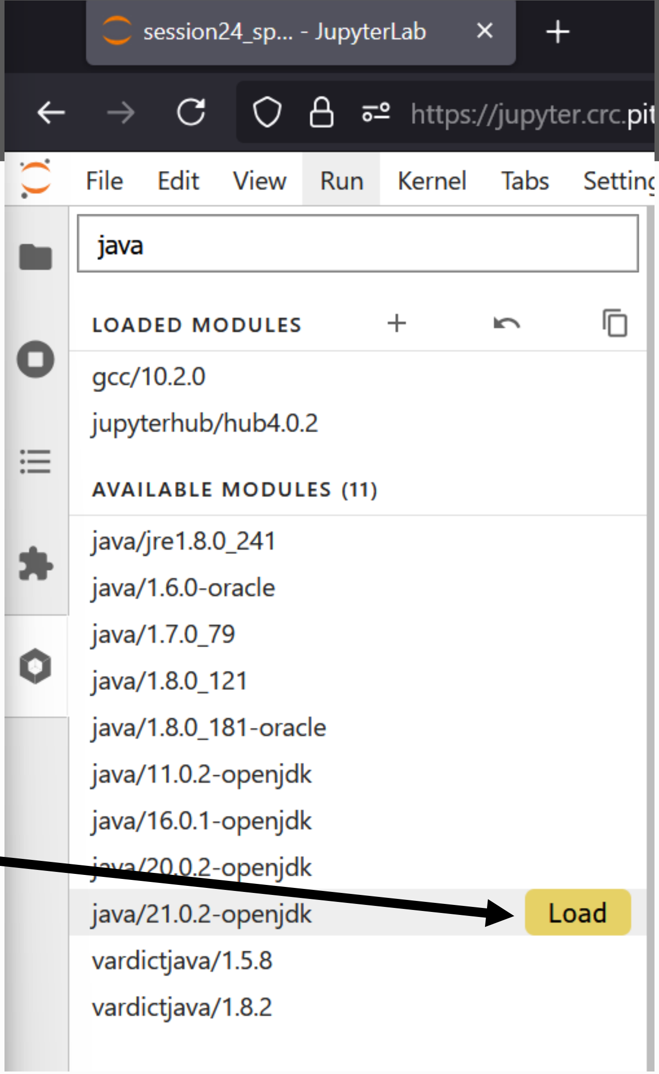
- Click on this nbgitpuller link or find the link on the course website
- Start a regular CPU 'Teach – 6 cores, 3 hours' server. There is no need for a GPU



- No need to load any notebooks yet

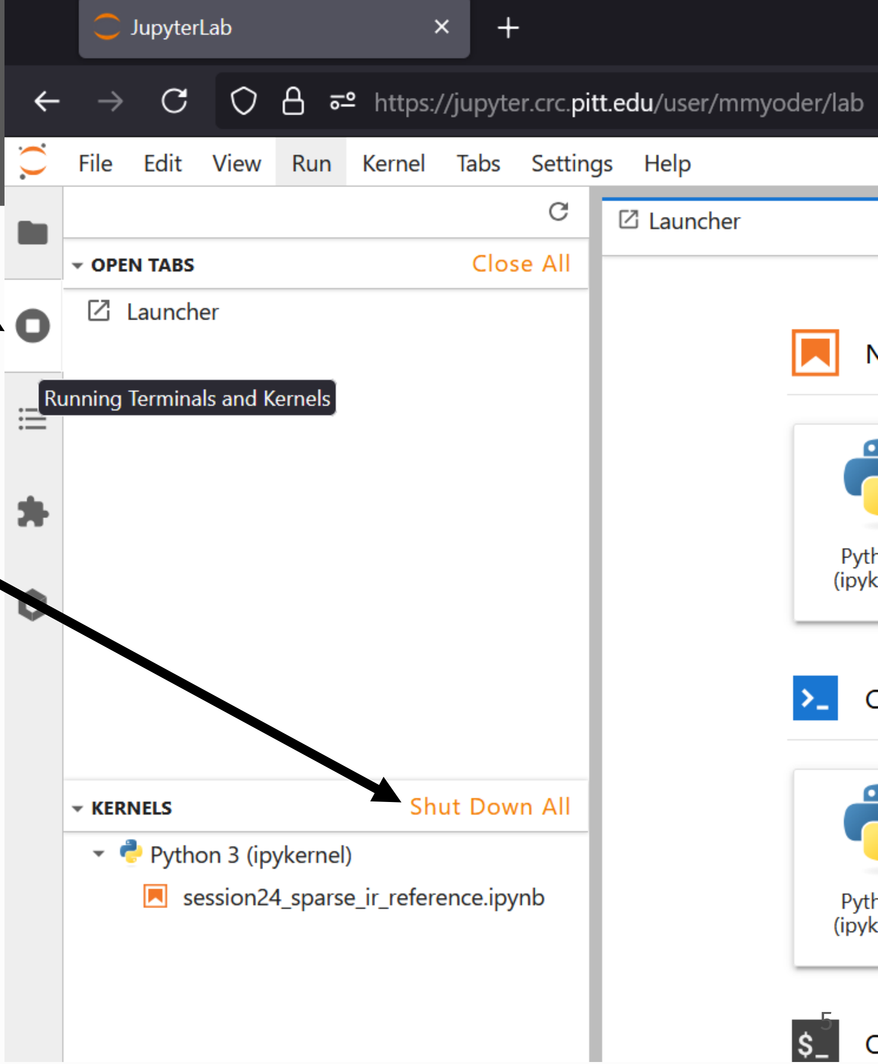
Load Java module

- The pyserini package requires a specific version of the Java JDK. We will be loading it as a module through JupyterHub
- Click the Software Modules icon the left-hand sidebar
- Filter for “java”
- Click **Load** next to **java/21.0.2.-openjdk**
    - Open **session24_sparse_ir.ipynb**



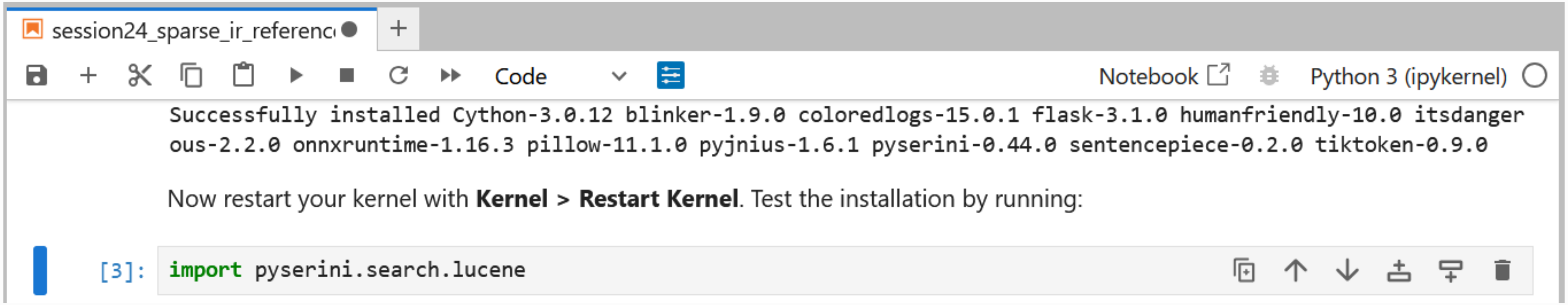
Make sure all kernels are shut down

- Click the 2nd icon down on the sidebar to view any kernels that are running
- Click **Shut Down All** if you any kernels are open
- Then launch **session24_sparse_ir.ipynb**



Load pyserini package

- Run **session24_sparse_ir.ipynb** through the following cell, which will take a long time to run the first time:



Learning objectives: information retrieval (IR), RAG

Students will be able to:

- Diagram the process of **classic information retrieval based on sparse embeddings**
- Describe how **retrieval-augmented generation (RAG)** works
- List software that can be used to build classic IR systems and RAG
- Identify and explain a common evaluation IR evaluation metric, **mean reciprocal rank (MRR)**

Information retrieval (search)

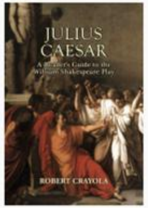
Information retrieval and question answering

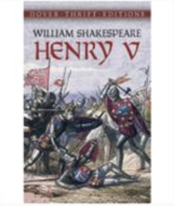
- Information retrieval (IR)
  - Choosing the most relevant document/s from a set of documents given a user's query
  - Search engines
- Closely related to question answering (QA)

Traditional IR: sparse embeddings

Sparse embeddings (bag-of-words) of documents and queries

- Each cell is the count of term t in a document d ($tf_{t,d}$).
- Each document is a count vector in $\mathbb{N}^V$, a column below.





	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	1	8	15
soldier	2	2	12	36
fool	37	58	1	5
clown	6	117	0	0

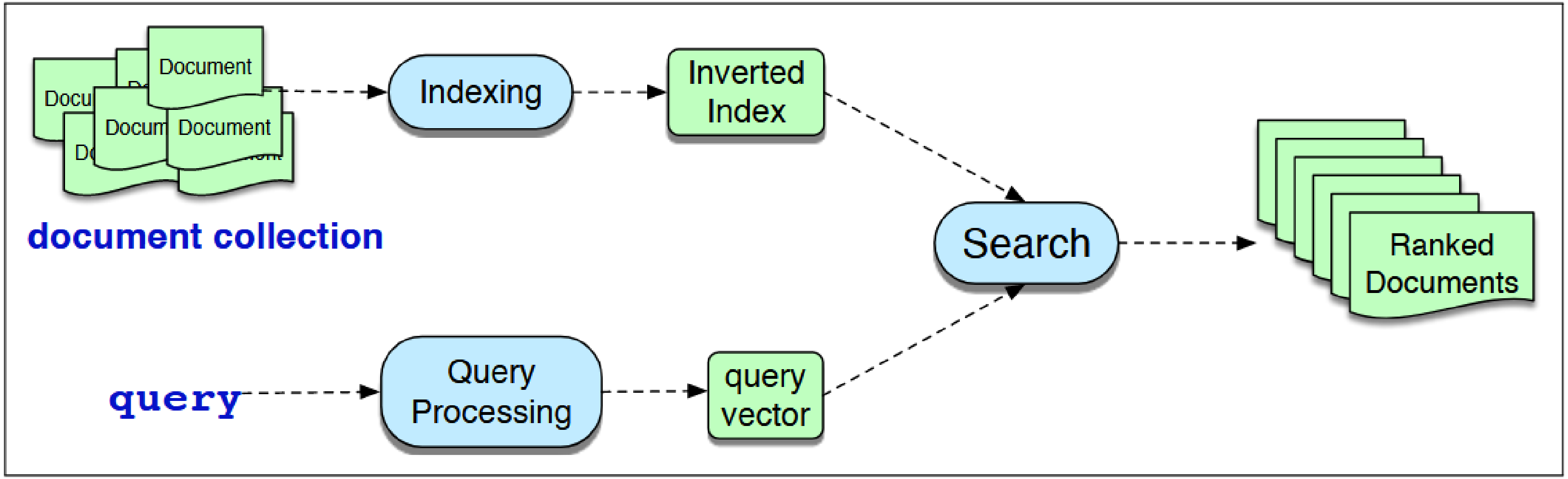
Slide based on David Mortensen, Jurafsky & Martin

BM25 transformations of bag-of-word vectors

- Modification of tf-idf
- Additional parameters:
  - *k* to control how much we care about word frequency
  - *b* to control how much we care about document length normalization
- Score of document *d* given query *q:*

$$\sum_{t\in q} \overbrace{\log\left(\frac{N}{df_t}\right)}^{\text{IDF}} \overbrace{\frac{tf_{t,d}}{k\left(1-b+b\left(\frac{|d|}{|d_{\text{avg}}|}\right)\right)+tf_{t,d}}}^{\text{weighted tf}}$$

Traditional IR pipeline



- Return documents with most similar vectors to query vector (by cosine similarity)
- Inverted index: *term {document frequency} -> document_id1 [term frequency] document_id2 [term frequency]*
  - E.g. chicken {50} -> 774 [20] 32 [2]

Retrieval-augmented generation (RAG)

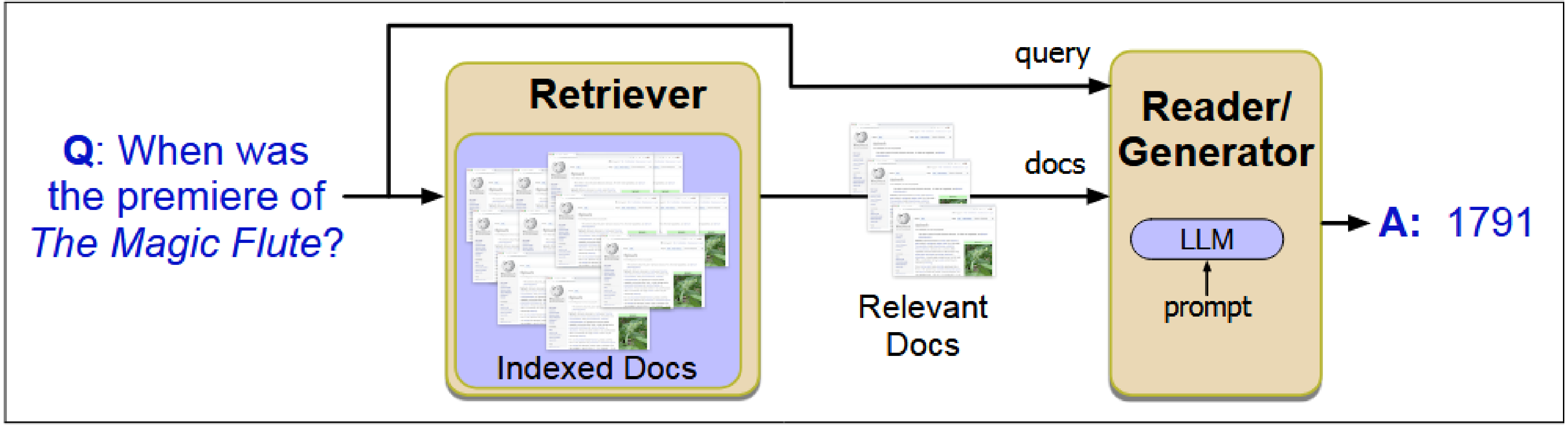
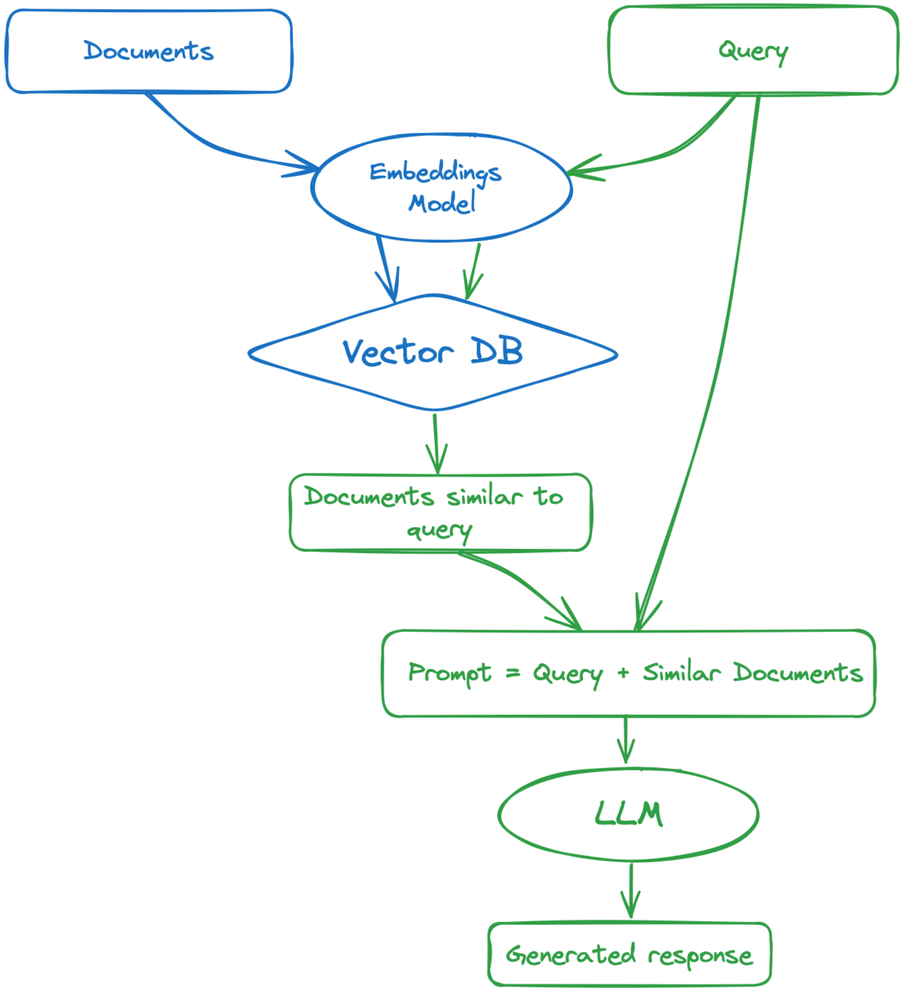


Figure 14.9 Retrieval-based question answering has two stages: **retrieval**, which returns relevant documents from the collection, and **reading**, in which an LLM **generates** answers given the documents as a prompt.



Credit: Maria Khalusova
https://huggingface.co/learn/cookbook/en/rag_zephyr_langchain

Coding activity

Notebooks to explore

- session24_sparse_ir.ipynb
  - Record:
    - Observations from trying different queries on MS MARCO
    - Mean reciprocal rank (MRR) on MS MARCO dev subset
- session24_rag.ipynb
  - Record:
    - Comparison between directly asking LLM and doing RAG
- If you finish early, try building a classic IR or RAG system on a new corpus of your choosing!

Wrapping up

- Classic information retrieval returns documents based on cosine similarity to the query's sparse embeddings, often transformed with tf-idf or BM25
- Retrieval-augmented generation provides relevant documents as context to an LLM to generate a response to prompts and questions
- Mean reciprocal rank (MRR) can be used for evaluation of information retrieval systems

Questions?