# CS 2731
# Introduction to Natural Language Processing

## Session 25: Information retrieval, RAG

Michael Miller Yoder

December 2, 2024


School of Computing and Information

# Course logistics

- Next class session this Wed Dec 4 will be open project work time
  - I'll be available to help assist groups
- **No class next Mon Dec 9**
- Final project presentations are **next Wed Dec 11**
- Project report is **due next Thu Dec 12**

# 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

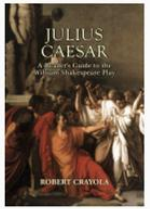
- Information retrieval (IR)
  - Choosing the most relevant document/s from a set of documents given a user's query
  - Search engines
- Closely related to question answering (QA)

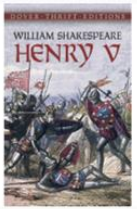
# Traditional IR: sparse embeddings

## 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 |

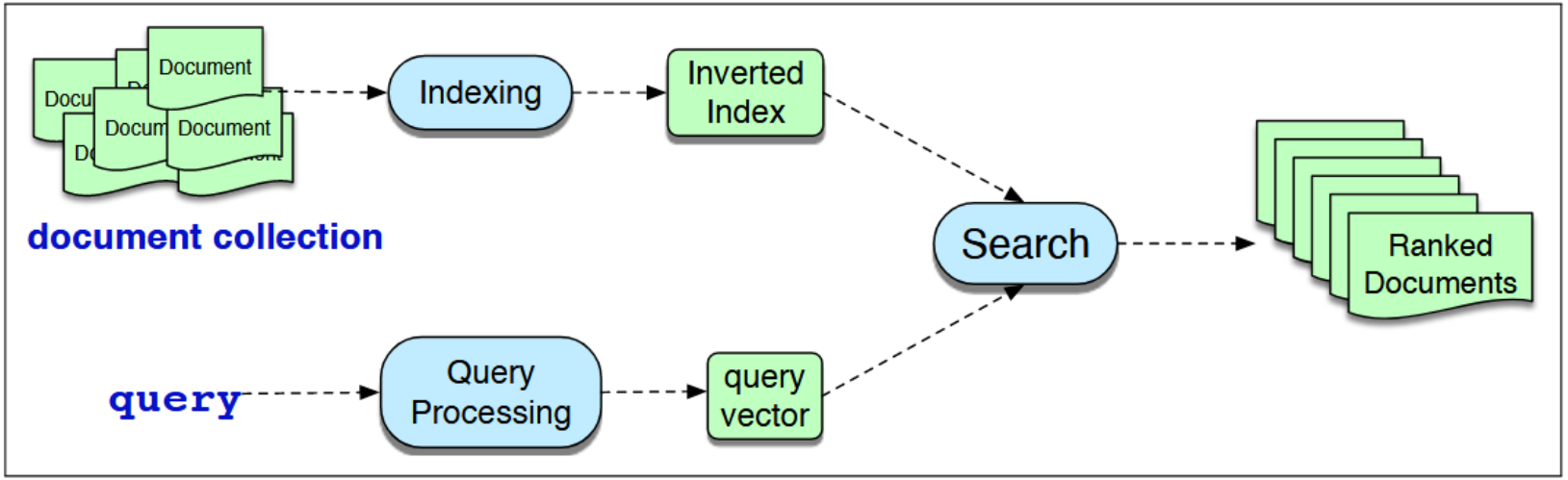
*Slide based on David Mortensen, Jurafsky & Martin*

# 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)

# Hands-on coding activity

# Notebooks to explore

- sparse_ir_skeleton.ipynb
  - Run on CPU
  - Record:
    - Observations from trying different queries on MS MARCO
    - Mean reciprocal rank (MRR) on MS MARCO dev subset
- rag_skeleton.ipynb
  - Run on GPU
  - 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!

# Retrieval-augmented generation (RAG)

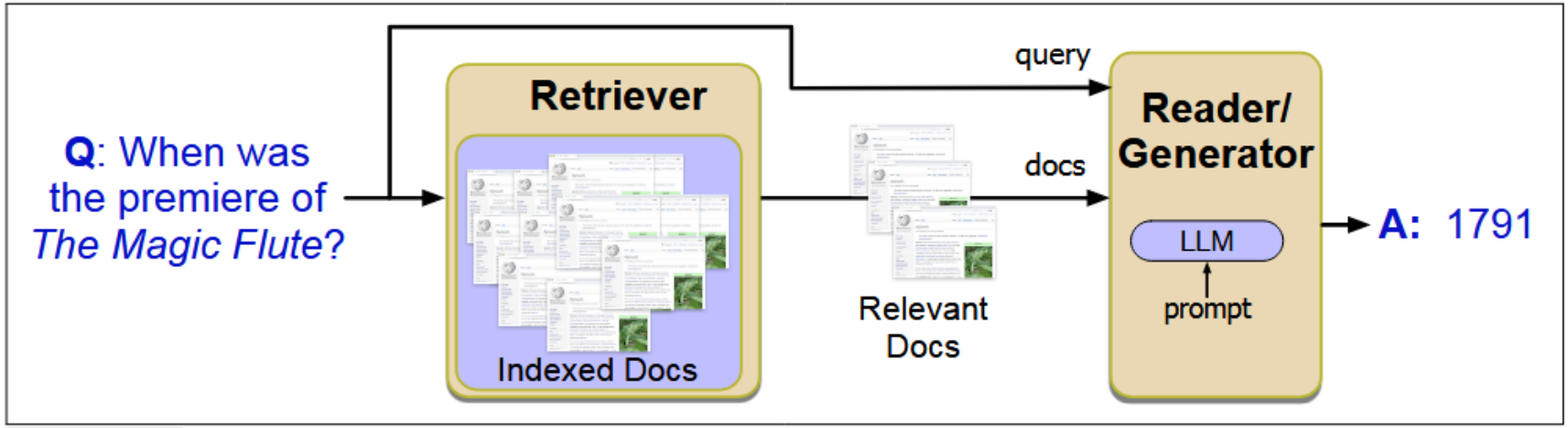
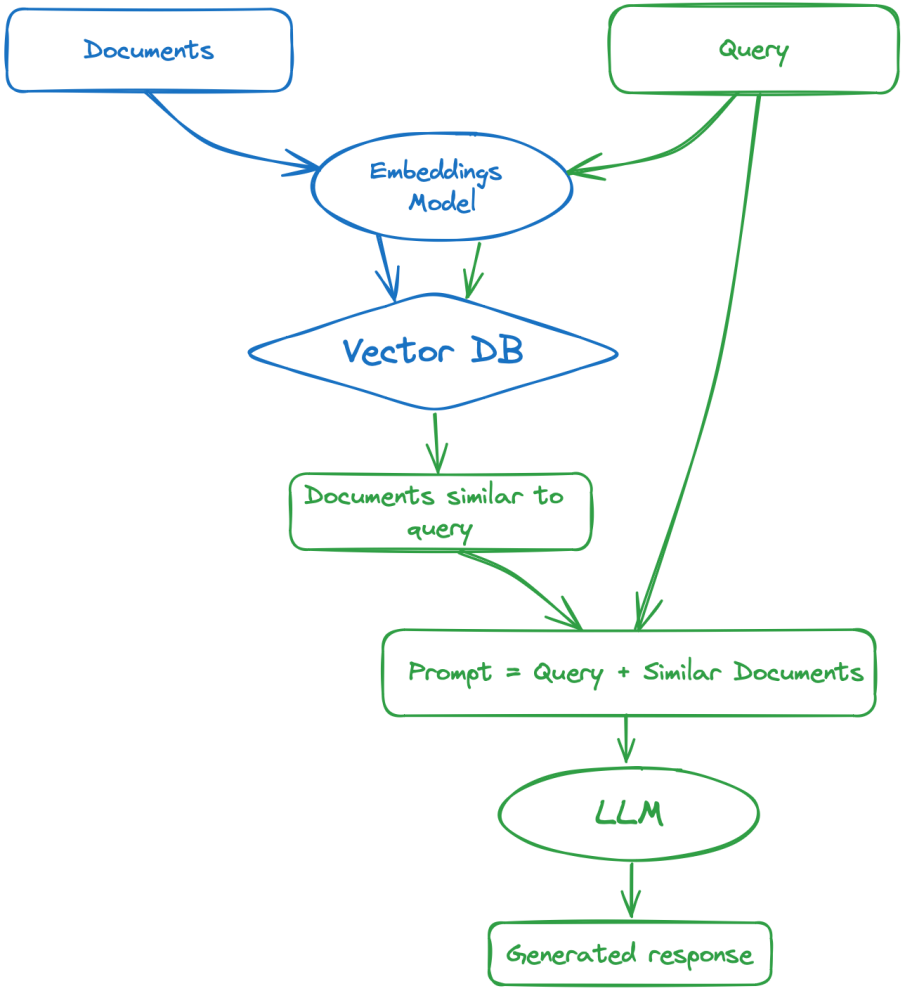



**Figure 14.9** Retrieval-based question answering has two stages: **retrieval**, which returns relevant documents from the collection, and **reading**, in which an LLM **generates** answers given the documents as a prompt.




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

# 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?*