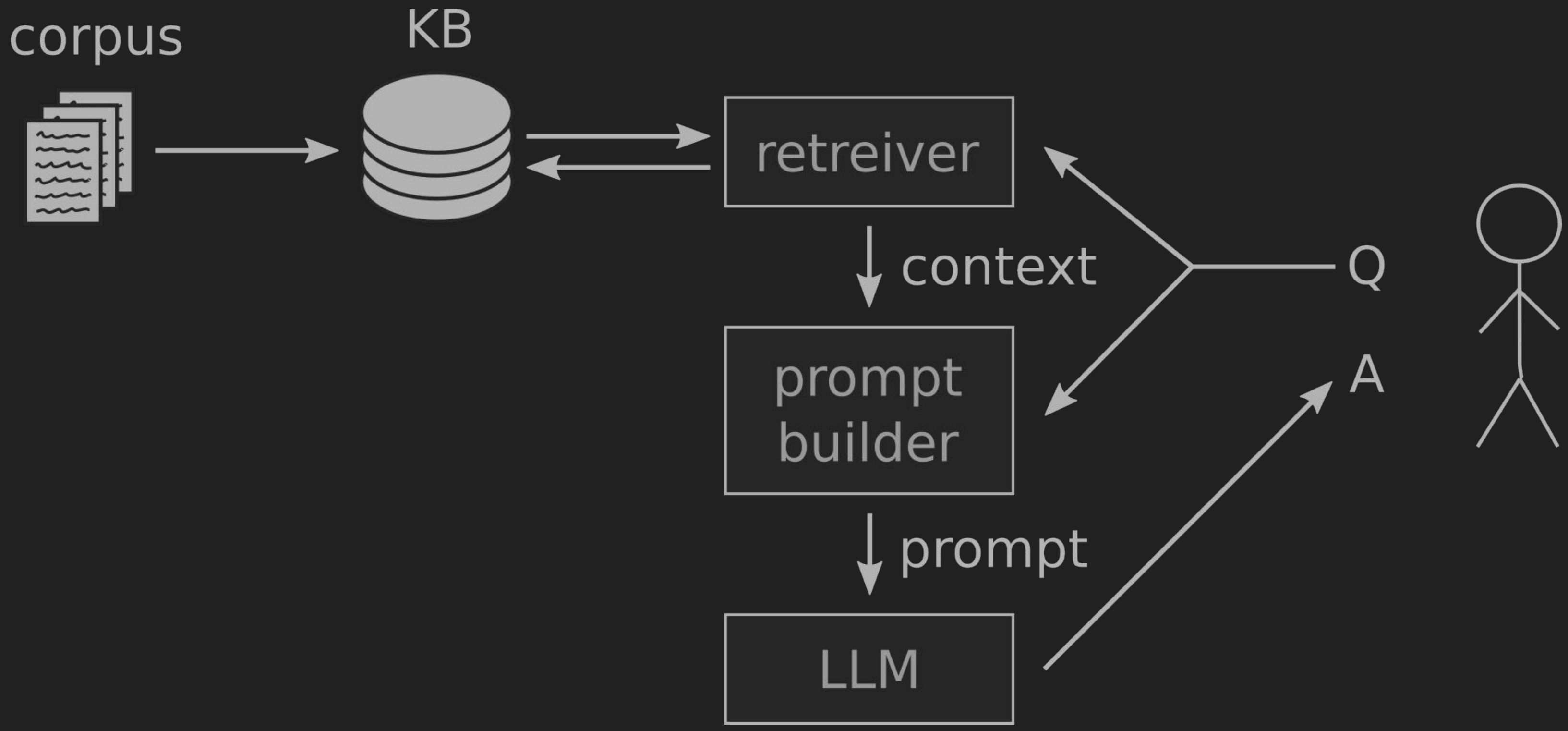
Challenges in building RAGs

Robert Różański
yhteys.ai

Why Retrieval Augmented Generation?

- issues with using LLMs for QA:
  - hallucinations
  - sources
  - unseen information (new / domain specific / proprietary)
- potential solution:
  - provide context: information necessary to answer
  - size limit - can’t just dump documents on LLMs
- RAGs:
  - first form context by retrieving information relevant for the question
  - use LLM to answer to generate an answer from the context and the question


corpus
KB
retreiver
context
prompt
builder
prompt
LLM
Q
A

Questions are not made equal

- Zhao et al. - stratification of questions
  - 1: explicit facts
    - “Where will the 2024 Summer Olympics be held?”
  - 2: implicit facts
    - “What is the majority party now in the country where Canberra is located?”
  - 3: interpretable rationales: apply domain-specific rationales integral to the data’s context
    - E.g. diagnostics questions (answers need to follow FDA guides or local equivalent)
  - 4: hidden rationales: the rationales are not explicitly documented
    - Addressing security incidents in IT context (rationales implicit in past response pattern)
- Evaluation datasets cover only 1 and 2
- Anthropomorphisation makes managing expectations difficult

What to do with the questions?

- Unchanged, straight to retrieval
- Transformations (LLM)
  - Query rewrite
    - reformulation (cleanup)
    - step-back prompting (more general question→apply the general answer)
    - split query into multiple, elementary queries
  - Translation into a formal query language (DB use)
  - HyDE: generate potential text containing relevant info
- tradeoff: dev. time, complexity, cost, time vs performance

Retrieval

- one-off
  - go through the database and retrieve n-most relevant fragments
- multiple calls
  - e.g. when original question was re-written
- ordering the retrieved information:
  - ranking
  - re-ranking
    - score based: combine retrieved fragments from multiple calls (frequency=score)
    - use LLM to estimate similarity/relevance for the question
- tradeoff: dev. time, complexity, cost, time vs performance

Knowledge Base

- Vectors
  - sparse: most representative keywords
    - TF-IDF: keyword frequency vs reciprocal freq.
    - BM25: keyword counts vs document length and avg. doc. length
  - dense: text embeddings (cosine similarity)
    - BERT derivatives
    - LLM2vec
- Graphs
  - NLP, ontologies, information extraction
- Challenges
  - chunking (reindexing)
  - tradeoff: dev. time, complexity, cost, time vs performance

Preparing content for Knowledge Bases

- Loads of problems
- Multimodal
  - currently: text as the common language
  - audio: ASR
  - pictures: text embedding
  - video: ?
- PDFs: issues with extracting texts
- Embedded images
- Tables

More problems

- Evaluation
  - Overall:
    - LLMs?
  - Retrieval:
    - the easiest to evaluate
    - existing datasets for evaluation
      - questions?
      - performance generalization?
- Explainability
- Data security / Privacy
  - KB with access control
  - finetuning problematic
- Legal consequences
  - client-facing systems

Some references

- evaluation of RAGs: https://arxiv.org/abs/2405.07437
- RAG deepdive: https://arxiv.org/abs/2409.14924
- neo4j RAG builder:
  - https://neo4j.com/developer-blog/graphrag-llm-knowledge-graph-builder/
  - https://llm-graph-builder.neo4jlabs.com/