RETRIEVAL AUGMENTED GENERATION FOR ACADEMIC RESEARCH RELATED TO MACHINE LEARNING

Group: QueryMinds

MOHAMMADALI DEHGHANI

AMIR SAADATI

AMINA KADIC

MELIHA KASAPOVIC

MAIN GOALS

- Implementing a pipeline for answering 15 predefined questions related to machine learning and retrieval augmented generation.
- Exploring different retrieval approaches
- Implementing LangChain Framework
- Implementing Vanilla RAG
- Exploring hallucinations

CREATING CORPUS OF DOCUMENTS

- Documents downloaded from **arXiv API.**
- 100 documents retrieved with query: "retrieval augmented generation".
- Rest of the documents retrieved with query: "machine learning".
- Final Corpus has 191 pdf documents and their meta data.

TEXT EXTRACTION, SEGMENTATION & TOKENIZATION


```
{
  "title": [
    {
      "sentence": "Electre Tri-Machine Learning Approach to the Record Linkage Problem",
      "tokens": [
        "Electre",
        "Tri-Machine",
        "Learning",
        "Approach",
        "to",
        "the",
        "Record",
        "Linkage",
        "Problem"
      ]
    }
  ],
  "abstract": [
    {
      "sentence": "In this short paper, the Electre Tri-Machine Learning Method, generally
      "tokens": [
        "In",
```


- Converted raw PDFs into **structured text**
- Implemented two extraction pipelines:
  - **GROBID-based extraction** (primary pipeline)
  - **pdfminer-based extraction** (baseline for comparison)
- Extracted core scientific sections:
  - title, abstract, introduction, body text
- Cleaned extracted text from **noise and formatting artifacts**
- Performed **sentence segmentation** using nltk: sent tokenize
- Applied **word tokenization** using nltk: word tokenize
- Stored processed outputs as **JSON files**

NORMALIZATION, LEMMATIZATION & CONLL-U CONVERSION

- Text normalization:
  - **punctuation cleaning**
  - **lowercase conversion**
  - **spacing normalization**
- **Lemmatization and POS tagging** using Stanza
- Converted into **CoNLL-U format**
- Ensured standardized **10-column structure per sentence**

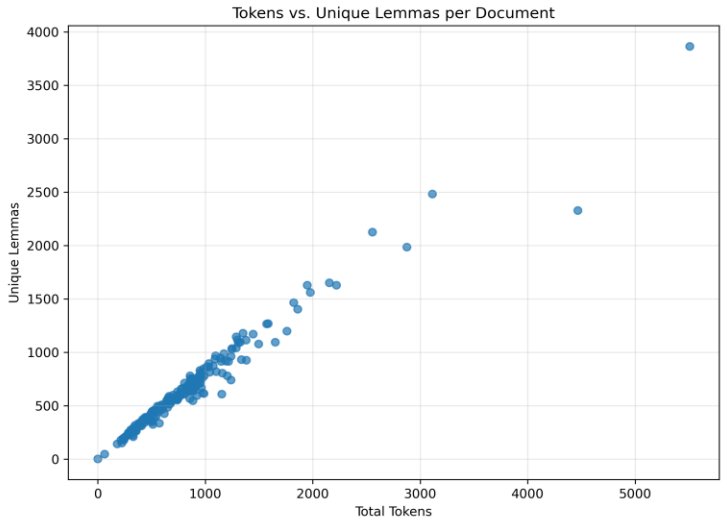
CONLL-U

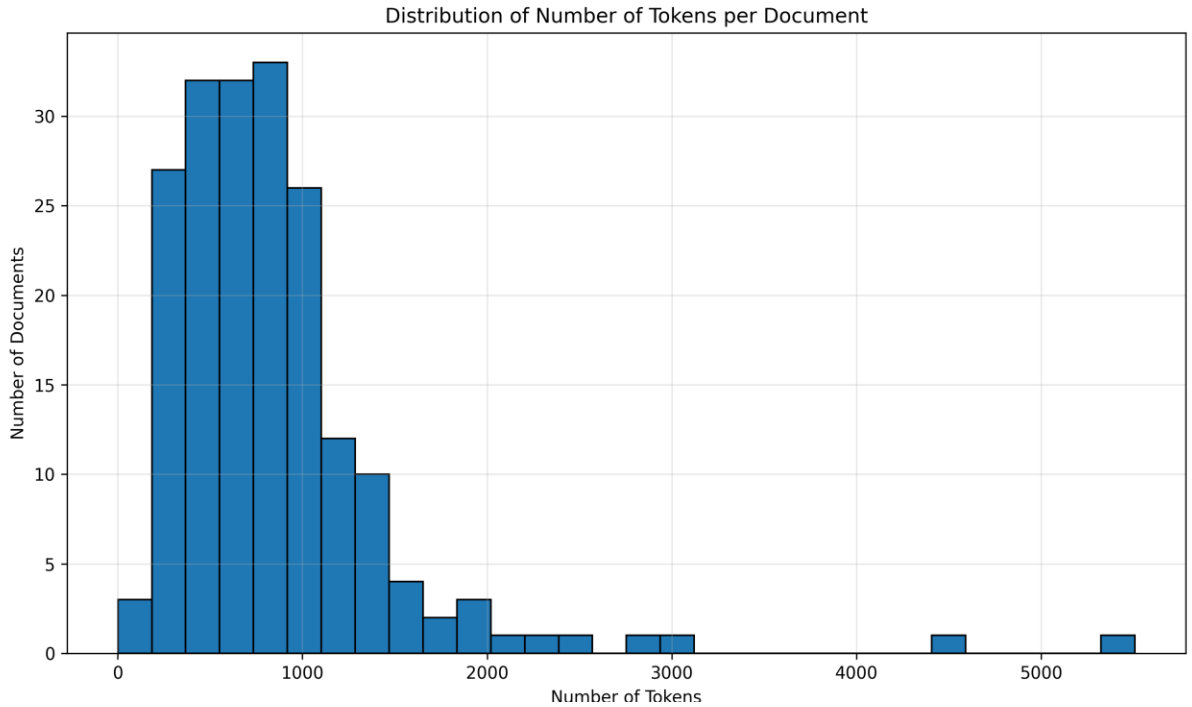
```
# text = an optimal control view of adversarial machine learning i describe an optimal control view of adversarial machine learning, where the dynamical system is th
input are adversarial actions, and the control costs are defined by the adversary's goals to do harm and be hard to detect.
# sent_id = 0
1       an      a       DET     DT      Definite=Ind|PronType=Art       0       _       _       start_char=0|end_char=2
2       optimal optimal ADJ     JJ      Degree=Pos      1       _       _       start_char=3|end_char=10
3       control control NOUN    NN      Number=Sing     2       _       _       start_char=11|end_char=18
4       view    view    NOUN    NN      Number=Sing     3       _       _       start_char=19|end_char=23
5       of      of      ADP     IN      _       4       _       _       start_char=24|end_char=26
6       adversarial     adversarial     ADJ     JJ      Degree=Pos      5       _       _       start_char=27|end_char=38
7       machine machine NOUN    NN      Number=Sing     6       _       _       start_char=39|end_char=46
8       learning        learning        NOUN    NN      Number=Sing     7       _       _       start_char=47|end_char=55
9       i       I       PRON    PRP     Case=Nom|Number=Sing|Person=1|PronType=Prs      8       _       _       start_char=56|end_char=57
10      describe        describe        VERB    VBP     Mood=Ind|Number=Sing|Person=1|Tense=Pres|VerbForm=Fin   9       _       _       start_char=58|end_char=66
11      an      a       DET     DT      Definite=Ind|PronType=Art       10      _       _       start_char=67|end_char=69
12      optimal optimal ADJ     JJ      Degree=Pos      11      _       _       start_char=70|end_char=77
13      control control NOUN    NN      Number=Sing     12      _       _       start_char=78|end_char=85
14      view    view    NOUN    NN      Number=Sing     13      _       _       start_char=86|end_char=90
15      of      of      ADP     IN      _       14      _       _       start_char=91|end_char=93
16      adversarial     adversarial     ADJ     JJ      Degree=Pos      15      _       _       start_char=94|end_char=105
17      machine machine NOUN    NN      Number=Sing     16      _       _       start_char=106|end_char=113
18      learning        learning        NOUN    NN      Number=Sing     17      _       _       SpaceAfter=No|start_char=114|end_char=122
```

QUALITY CONTROL, STATISTICS & DOCUMENTATION

- Verified correctness of structure, lemmas, POS tags
- Calculated corpus statistics

Statistic	Sentences	Tokens	Unique Lemmas	Tokens per Sentence
Count	191	191	191	191
Mean	25.52	853.83	664.82	33.93
Std	16.43	643.87	453.41	10.14
Min	1	1	1	1
25%	15	489.5	381	27.26
50%	22	739	573	32.20
75%	31	985	786.5	39.27
Max	107	5507	3865	82.43





CORPUS STATISTICS

QUESTIONS AND LABELING


```
{"question_id": "q1", "chunk_id": "1501.04309v1_abstract_0001", "label": 1}
{"question_id": "q1", "chunk_id": "1501.04309v1_introduction_0003", "label": 1}
{"question_id": "q1", "chunk_id": "1504.03874v1_abstract_0001", "label": 0}
{"question_id": "q1", "chunk_id": "1504.03874v1_introduction_0003", "label": 0}
{"question_id": "q1", "chunk_id": "1504.03874v1_introduction_0005", "label": 0}
{"question_id": "q1", "chunk_id": "1505.06614v1_title_0000", "label": 1}
{"question_id": "q1", "chunk_id": "1505.06614v1_abstract_0001", "label": 1}
{"question_id": "q1", "chunk_id": "1505.06614v1_introduction_0003", "label": 1}
{"question_id": "q1", "chunk_id": "1505.06614v1_introduction_0004", "label": 1}
{"question_id": "q1", "chunk_id": "1505.06614v1_introduction_0005", "label": 1}
{"question_id": "q1", "chunk_id": "1505.06614v1_introduction_0006", "label": 1}
{"question_id": "q1", "chunk_id": "1505.06614v1_introduction_0007", "label": 1}
{"question_id": "q1", "chunk_id": "1505.06614v1_introduction_0010", "label": 1}
{"question_id": "q1", "chunk_id": "1612.04858v1_abstract_0001", "label": 1}
{"question_id": "q1", "chunk_id": "1612.04858v1_introduction_0002", "label": 1}
{"question_id": "q1", "chunk_id": "1612.07640v1_introduction_0007", "label": 1}
{"question_id": "q1", "chunk_id": "1612.07640v1_introduction_0008", "label": 1}
{"question_id": "q1", "chunk_id": "1612.07640v1_introduction_0010", "label": 1}
{"question_id": "q1", "chunk_id": "1703.10121v1_introduction_0002", "label": 1}
{"question_id": "q1", "chunk_id": "1706.05749v1_abstract_0001", "label": 1}
{"question_id": "q2", "chunk_id": "1501.04309v1_introduction_0003", "label": 1}
{"question_id": "q2", "chunk_id": "1706.05749v1_introduction_0004", "label": 1}
```


- 15 questions.
- Chunks were retrieved from 30 randomly picked documents.
- Each chunk has 200 tokens with the last 50 tokens overlapping with the following chunk.
- Keyword check performed between questions and retrieved chunks to produce candidate chunks.
- Candidate chunks were hand labeled and saved as gold labels (274 gold labels).

RULE-BASED BASELINES

- Built a **keyword overlap baseline**:
  - scores chunks based on shared terms with the question
- Implemented **TF-IDF + cosine similarity** baseline
- Used baselines for:
  - ranking chunks
  - binary relevance classification
- **Evaluation script** computing:
  - Accuracy, Precision, Recall, F1
  - Precision@k and Recall@k

ML BASELINES

- **Embedding-based retrieval -** SentenceTransformers MiniLM-L6-v2
- **Supervised classifier** - TF–IDF + logistic regression
- Analyzed **false positives and false negatives**

EMBEDDING BASED RETRIEVAL EXAMPLES

- **Question:**
  "Which datasets or types of data are used in the experiments, and for what reasons are they chosen?"
- **Retrieved Chunk (label = 1):**

"It is well known that ML algorithms are affected by the curse of dimensionality [11] , but ML practitioners also know that it could be possible to obtain reliable models even for high-dimensional data sets , and with a relatively small number of samples [12] ...",

- **Question:**
  “How are concepts from information theory connected to learning targets?”
- **Retrieved Chunk (label = 0):**

“Information Theory and its Relation to Machine Learning” (title)

SUPERVISED CLASSIFIERS EXAMPLES

- **Question:**
  "In what ways are machine learning methods applied to concrete scientific or engineering problems?"
- **Retrieved Chunk (label = 1):**

"Introduction Recently , there has been interest in applying Bayesian black-box optimization strategies to better conduct optimization over hyperparameter configurations of machine learning models and systems...",

- **Retrieved Chunk (label = 0):**

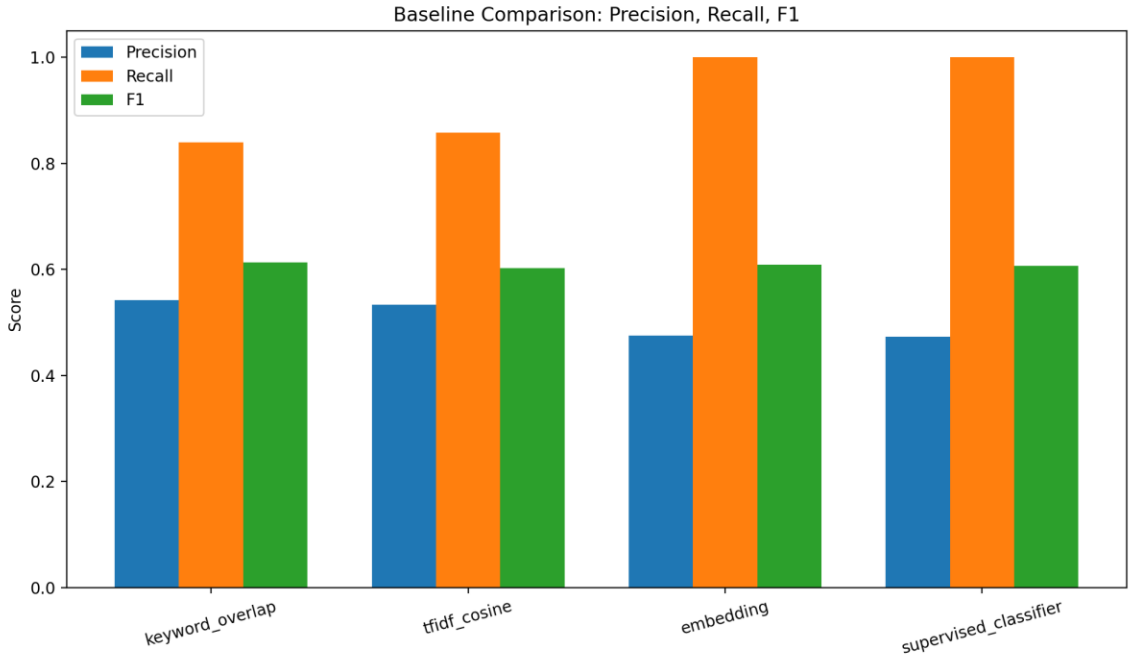
"when: the programme becomes better at the task at hand ; the programme can perform the task more efficiently ; the code becomes `` more structured '' and simpler . One possible analogy to better understand the above statements can be found in software engineering . When considering code that performs a specific task , we do not care only about its functionality , but also about its executio..."

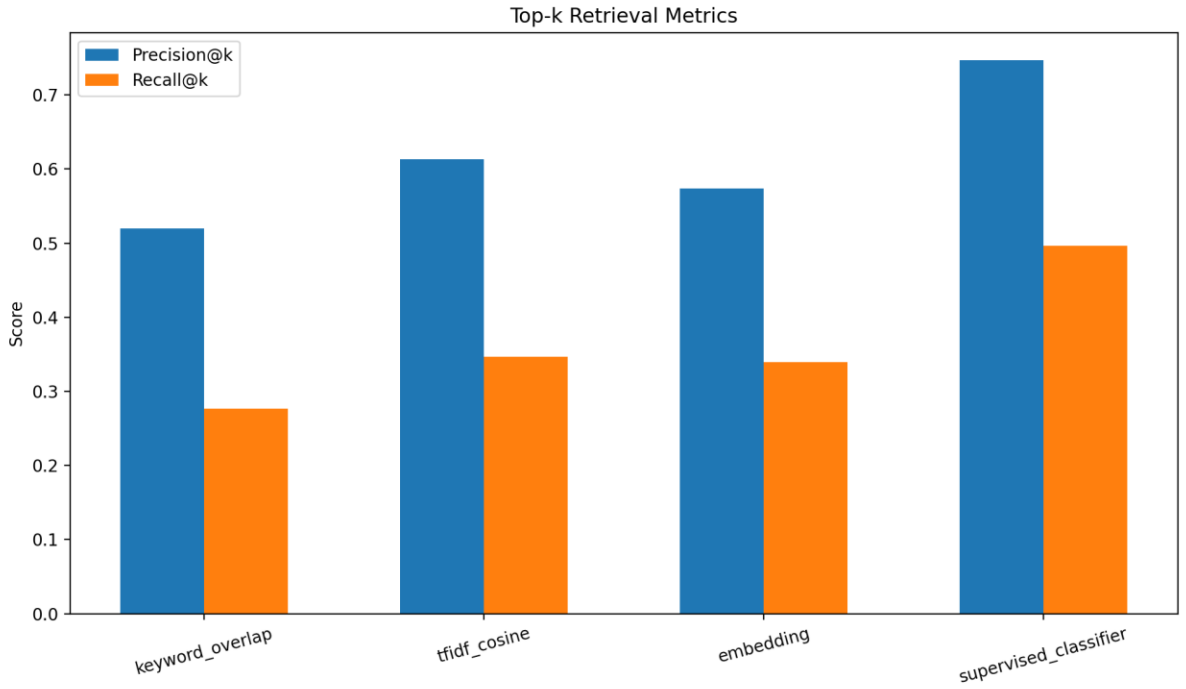
INTEGRATION & COMPARISON

- Implemented a **unified pipeline which runs all baselines**
- Implemented **baseline comparison scripts**
- Generated **comparison tables and plots**

Baseline	Precision	Recall	F1	Accuracy	Precision@5	Recall@5
Keyword Overlap	0.5421	0.8396	0.6131	0.5474	0.5200	0.2767
TF-IDF Cosine	0.5331	0.8574	0.6020	0.5615	0.6133	0.3465
Embedding	0.4751	1.0000	0.6090	0.4793	0.5733	0.3392
Supervised	0.4727	1.0000	0.6065	0.4727	0.7467	0.4967

BASELINES COMPARISONS





VANILLA RAG

```
prompt = PromptTemplate(
    input_variables=["context", "question"],
    template="""
You are an AI assistant.

Use the following document excerpts to answer the

Context:
{context}

Question:
{question}

Answer in a clear and concise way.
"""
)
```

- Generated **dense embeddings** using `all-MiniLM-L6-v2`
- Built and stored a **FAISS vector index** for efficient similarity search
- Retrieved **top-k = 5** most relevant chunks per question
- Designed a **basic (unconstrained) RAG prompt**
- Implemented **Vanilla RAG pipeline** using LangChain
- Generated answers for all questions defined with TinyLlama/TinyLlama-1.1B-Chat-v1.0
- Identified and collected **typical hallucination examples**

EXAMPLES

- **Example 1**
- **Question ID:** q8
  Question: "In what ways are machine learning methods applied to concrete scientific or engineering problems?"

- **Answer:** Image classification: Machine learning algorithms can be used to recognize objects in an image. For instance, in image recognition, there are several classification tasks, such as object detection, object segmentation, and object recognition....

- **Observed hallucination:** The retreived chunks mention image recongition, but don't talk about it in detail.

- **Example 2**
- **Question ID:** q14
  Question: "How are machine learning models connected to real-world decision-making or application-oriented requirements?"

- **Answer:** A machine learning (ML) model is a mathematical function that can be trained to make predictions on new data. A machine learning model is created by feeding large amounts of data into a set of algorithms, which then use these algorithms to learn from the training data and generate predictions...

- **Observed hallucination:** Uses buzzwords to make it sound like it is answering the question, but never answers concretely.

CHUNK QUALITY & DATA IMPROVEMENTS

- Reformulate questions to more specific terms and clearer meaning
- Retrieve new chunks
- Label based on majority voting (TBD)

```
"question": "Which datasets or types of data are used in the experiments, and for what reasons are they chosen?",
```

⇨

```
"question": "What types of datasets or data sources are used for experiments, and what are typical reasons for choosing them?",
```

```
"question": "What forms of explanation or interpretability of model behavior are discussed, if any?"
```

⇨

```
"question": "What methods are used to explain or interpret model behavior (e.g., feature importance, saliency, counterfactual explanations)?"
```

CHANGES BETWEEN VANILLA VERSIONS

- Better chunking (smaller and more focused text units)
- Multi-annotator labeling (more reliable gold data)
- More document coverage and cleaner data
- Improved FAISS retrieval built on better chunks

VANILLA RAG COMPARISONS

System	Version	P@5	R@5 (within top 20)
Vanilla	v1	0.093	0.144
Vanilla	v2	0.267	0.205
ΔVanilla	v2 - v1	0.174	0.061

- Vanilla v2 retrieves more relevant chunks in the top-5 results.
- This directly improves answer quality because the language model sees better evidence.
- P@5 increased almost three times
- R@5 also increased - more correct information appears early

WHAT CHANGED AFTER BETTER CHUNKING

- What changed from v1 to v2:
- Better chunking (smaller and more focused text units)
- Multi-annotator labeling (more reliable gold data)
- More document coverage and cleaner data
- Improved FAISS retrieval built on better chunks

PLANNED IMPROVEMENTS

- Hallucination Detection
- Manual Labeling
- Constrained RAG Improvement

THANK YOU FOR YOUR ATTENTION

QUERYMINDS