# Scalable Language Technologies Lab

Summer Semester 2025

Prof. Dr. Martin Potthast, Lukas Gienapp

# Course Information

## Instructors

Prof. Dr. Martin Potthast

Lukas Gienapp

## Contact

- Contact: lukas.gienapp@uni-kassel.de
- Office hours: by appointment (online)
- Web: https://temir.org > Teaching > Scalable Language Technologies Lab

# Course Information

## Organization

- **Workload:** 4 SWS
- **Schedule:** Monday, 10:15-13:45
  - First 3 weeks: online class for all
  - From week 4 onwards: individual online group supervision meetings
- **Location:**
  - First meeting: Today, hybrid
  - Following meetings: Online, BBB
- **Communication:** Email, Discord
- **Materials:** Slides, papers, and resources on course website

# Scalable Language Technologies

Language technologies are methods and tools for analyzing, modifying, and generating human language.

- Support interactions between humans and machines in natural language
- Form the foundation of numerous intelligent information systems:
  - Search engines
  - Translation systems
  - Dialog and conversation systems
  - AI Agents
  - Argumentation systems
  - ...
- Research subjects of NLP and IR fields
- Rely on AI, ML, and especially Deep Learning techniques

**In this semester:** focus on ‘hot topic’ retrieval-augmented generation (RAG)

# Learning Objectives

- Work in a structured and self-supervised manner
- Apply current research in language technologies
- Develop and carry out experiments at scale
- Work with large corpora via state-of-the-art infrastructure
- Collaborate effectively in a group
- Scientific writing and presentation
- Create demonstrable software solutions

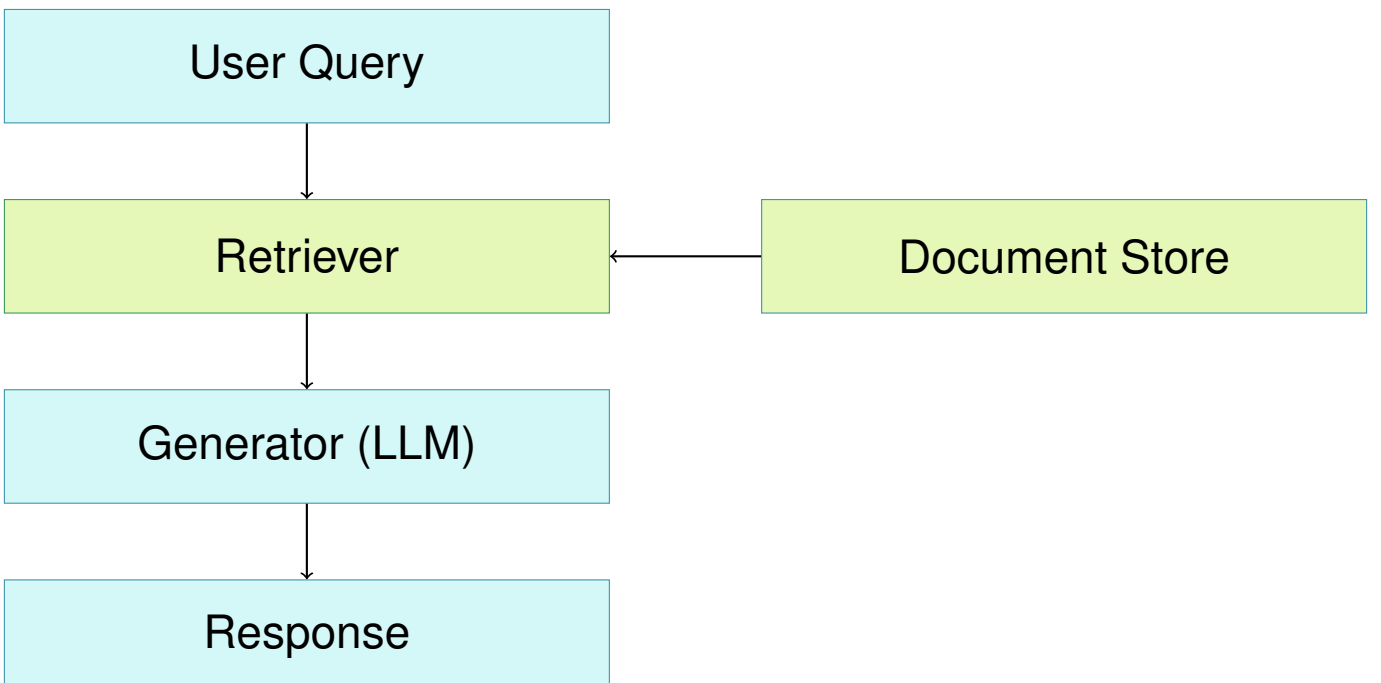
# Introduction to RAG

## What is Retrieval-Augmented Generation?

- Combination of two key approaches: **Retrieval** and **Generation**
  - **Retrieval:** Finding relevant information from a corpus
  - **Generation:** Creating coherent text with language models
  - ➜ Generation relies on context from retrieved sources to answer the query
- RAG addresses limitations of standalone LLMs
  - Knowledge cutoff
  - Hallucinations
  - Lack of specific domain knowledge
  - Non-verifiable claims
  - ...
- RAG addresses limitations of standalone Retrieval
  - Synthesizes from multiple sources
  - Provides direct answer (with some caveats...)
  - Can adjust to user preferences (language, accessibility, ...)
  - ...

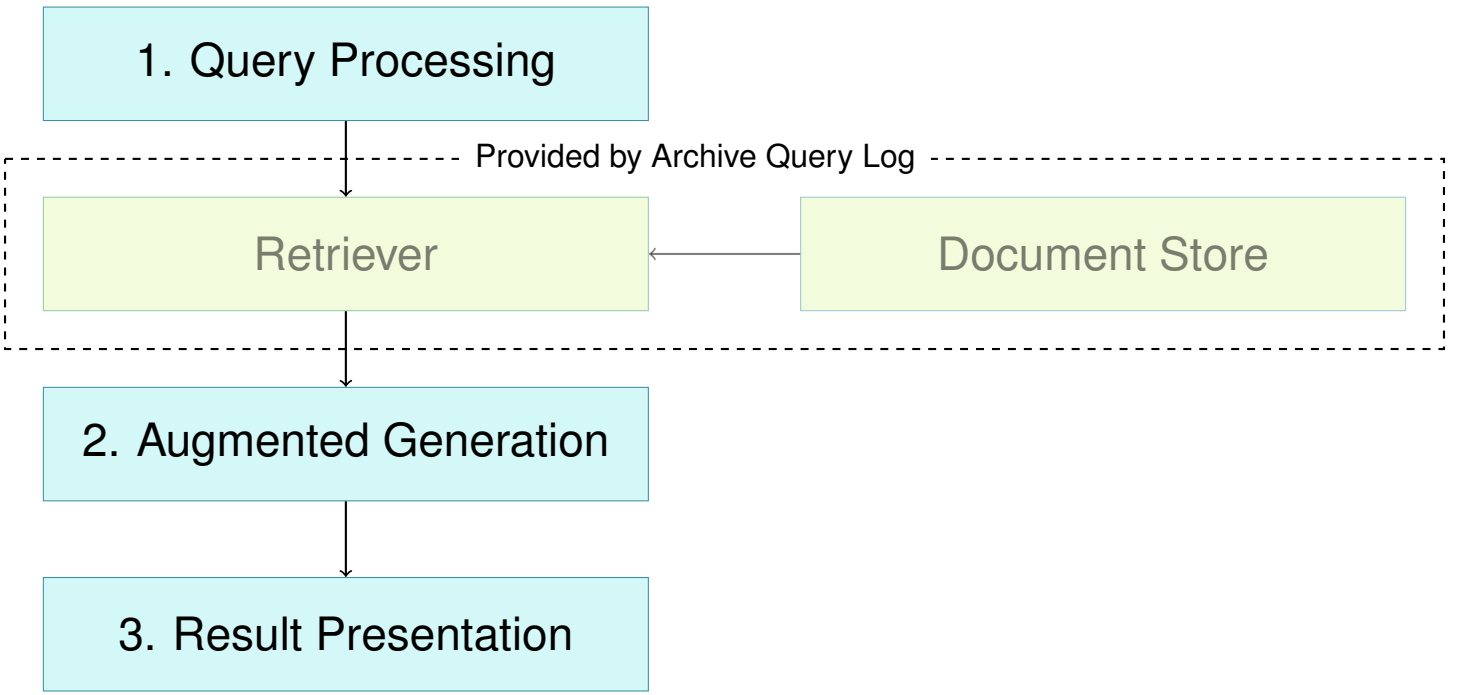
# Introduction to RAG

## How is a RAG system built?



# Introduction to RAG

The lab focuses on three key areas in RAG systems:

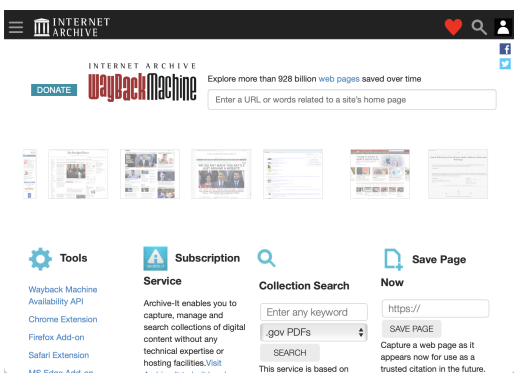



- **1. Query Processing:** Query analysis and suggestion systems
- **2. Augmented Generation:** RAG-based answer generation from snippets
- **3. Result Presentation:** SERP browsing and visualization

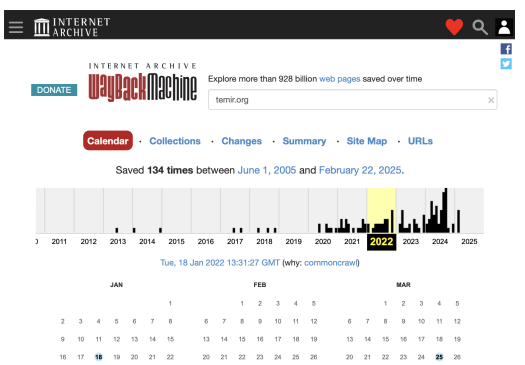
# Lab Resources: Archive Query Log

## Web Archive

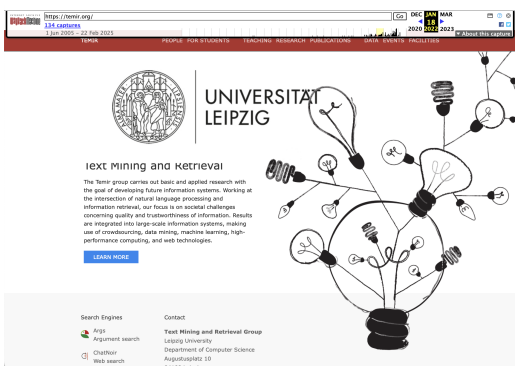
- archive.org is a non-profit aiming to publicly archive digital heritage
- Their web archive allows everyone to archive a website at any point in time
- These archived version can be retrieved and re-rendered




web.archive.org




Archives for temir.org



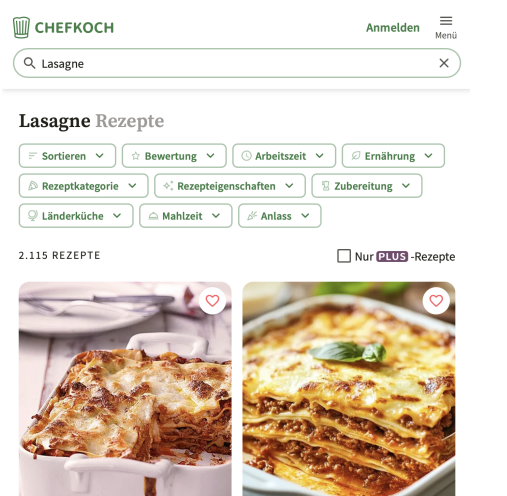

Single archived version

# Lab Resources: Archive Query Log

## Parsing Archived Search Pages

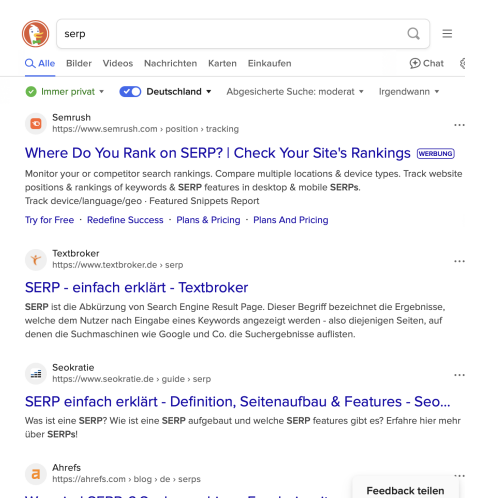
- AQL: archived SERPs in structured format for comprehensive query log
  - 356 million queries, many different information need types
  - 166 million search result pages (SERPs), spanning 25 years of web
  - 550 search providers, multiple search domains, multiple languages

➜ The AQL allows to simulate searches for RAG system development, by providing quick access to real web-scale retrieval results for research




Recipe Search


Video Search




Web Search

# Project Organization

- Work in small groups (3-5 students)
- Choose one of three project tracks:
  - **Project 1:** Query-Analytics & Suggestion-System
  - **Project 2:** RAG-based Answer Generation from SERP Snippets
  - **Project 3:** SERP-Browser & Visualization
  - ➜ Projects are a starting point – you can develop your own ideas with the scope of each project!
- All projects will use the AQL as a key resource
  - Access to raw data snapshots as JSONL dump for local experimentation
  - API-Access to full AQL data via an ElasticSearch cluster hosted by us
  - Access to a test query set from various domains as text file
- Projects should be demoable and well-documented

# Project 1: Query-Analytics and Suggestion-System

## Background

- This project focuses on the query processing component of the RAG pipeline
- Understanding how users formulate search queries is fundamental for improving search engines
- Provides insights into real query patterns across different search engines and time periods
- Makes the AQL data more accessible through query analytics and recommendations

## Expected Outcomes

- Comprehensive analysis of AQL query characteristics with self-chosen focus
- Interactive query suggestion system that can find similar queries in the AQL based on that analysis

# Project 1: Query-Analytics and Suggestion-System

## Tasks

- **Analyze queries and their results** in the AQL regarding, for example:
  - Search trends over time?
  - User intentions and information needs?
  - Query characteristics and complexity?
  - Patterns across different search engines?
- **Develop a query suggestion system** that:
  - Enables exploration of AQL queries, by making them searchable
  - Supports search by similarity, topic, or intent?
  - Helps understand how queries evolve over time?
  - ...

➜ **TL;DR:** given any query, suggest relevant ones from the AQL (defining what relevance means is up to you!)

# Project 2: RAG-based Answer Generation from SERP Snippets

## Background

- This project focuses on the generator component of the RAG pipeline
- Investigates if a RAG can transform fragmented snippets from the AQL into coherent, useful answers
- No access original documents - using only the snippets available in SERPs
- Challenges include handling diversity, inconsistency, and incompleteness of snippets

## Expected Outcomes

- Pipeline for snippet extraction and preprocessing, analysis of snippet characteristics
- Implementation of a RAG model using suitable LLMs (we can provide model & compute access)

# Project 2: RAG-based Answer Generation from SERP Snippets

## Tasks

- **Extract and analyse snippets** from the AQL:
  - Identify relevant snippets for different queries
  - Clean and normalize snippet content
  - Organize snippets for effective context provision
  - Gain insight into what information snippets usually offer in the AQL
- **Implement a RAG model** for answer generation:
  - Select appropriate LLMs for the task
  - Design effective prompting strategies
  - Handle evidence integration from multiple snippets
  - Adapt answer inference based on SERP characteristics (for example, develop domain-specific stragies, query intents, ...)

➜ **TL;DR:** given any AQL SERP, generate a good answer to its query based on its provided snippets (defining what good means is up to you!)

# Project 3: SERP-Browser

## Background

- This project focuses on the result display component of the RAG pipeline
- The AQL offers researchers a valuable tool for investigating search engines
- Current analysis is complex and requires technical expertise
- A web-based tool would make archived SERPs interactively accessible
- Particularly important for temporal and comparative analyses

## Expected Outcomes

- Full-stack web application for SERP browsing, with ElasticSearch as data provider (Vue.js + FastAPI recommended)
- Approach for exploration and visualization of SERPs

# Project 3: SERP-Browser

## Tasks

- **Develop a user interface** for browsing archived SERPs:
  - Intuitive navigation through the AQL data
  - Rendering of data in SERP layout, possibly with RAG answer
  - Filtering and sorting capabilities
- Implement **features to explore** the AQL database:
  - Search across different dimensions (queries, time, engines)
  - Extract insights from search results, for example through visualization

➜ **TL;DR:** develop a web app that allows to explore the AQL (how and what to explore is up to you!)

# Lab Deliverables

All projects require the following deliverables:

- **Report:** 10-page report in ACM format
  - What are you building?
  - How are you building it?
  - Why do you build it in that way?
- **Presentation:** 30-minute presentation + demo + questions
- **Software:** Code repository
- **Demo:** Working demonstration of your solution

Timeline:

- Project selection: End of Week 2
- Scheduled progress meetings: Weeks 5, 8, and 11
- Flexible consultations available in Weeks inbetween
- Final presentations: End of semester (exact data TBD.)
- Report submission: Two weeks after final presentations

# Getting Started

**Next Steps:**

- Form groups of 3-5 students
- Sign up for a project by the end of next week
- Schedule initial consultation with instructors
- Next Monday: exploring AQL using ElasticSearch (online)

**Resources:**

- Course materials will be available on the website
- Access to the AQL will be provided via ElasticSearch
- Computing resources available through Webis cluster
- Weekly online consultations available