scieneers
DRIVEN BY DATA

Who is an NLP expert? Lessons Learned from building an in-house QA-system

Alina Bickel, Nico Kreiling

scieneers
DRIVEN BY DATA

We gain knowledge from data and create value. For our customers, society and ourselves.

Alina Bickel

Working student @ scieneers

Nico Kreiling

Sr. Data Scientist @ scieneers

Host of techtiefen.de

How we got to this talk

scieneers
DRIVEN BY DATA

The crazy recent past of generativeAI

Yes, we also want to integrate ChatGPT-like models, but
No, we can not talk about this here yet.

ScieneersQA started

LangChain

ChatGPT

Scieneers QA release

Google Code Red

GPT-4

ChatGPT Plugin System

AutoGPT / BabyAGI

October November December January February March April

Some context on the project and this talk

scieneers
DRIVEN BY DATA

What is ScieneersQA

Develop a global **question-answer system** that can be easily used by any employee:

- With **no additional effort** for knowledge documentation
- Integration of **various data sources**
- Use of **existing communication systems**
- Collection of feedback for the purpose of further development

Things we will talk about

- Best practices for integrating heterogenous data sources into a QA system
- Evaluation of different retriever and reader systems
- Extractive vs. abstractive QA
- Leveraging a slack chatbot as user interface

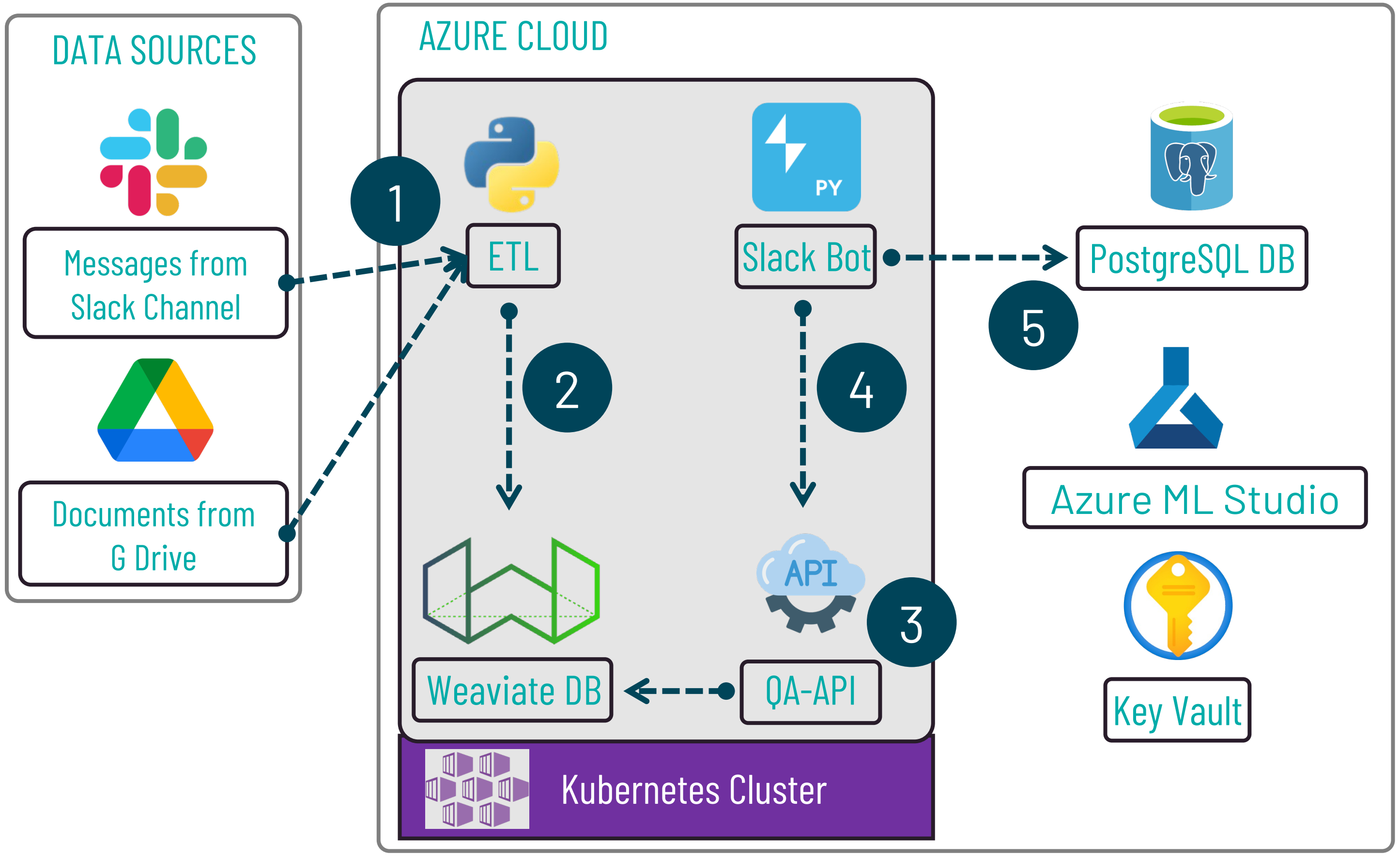
We won't talk about

- ChatGPT and RLHF-based LLMs

A first glimpse on the overall system

scieneers DRIVEN BY DATA

Architecture of the scieneers question-answer system



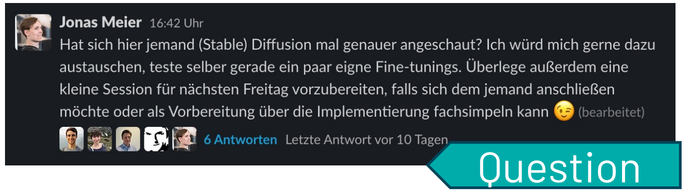
5 important processing steps

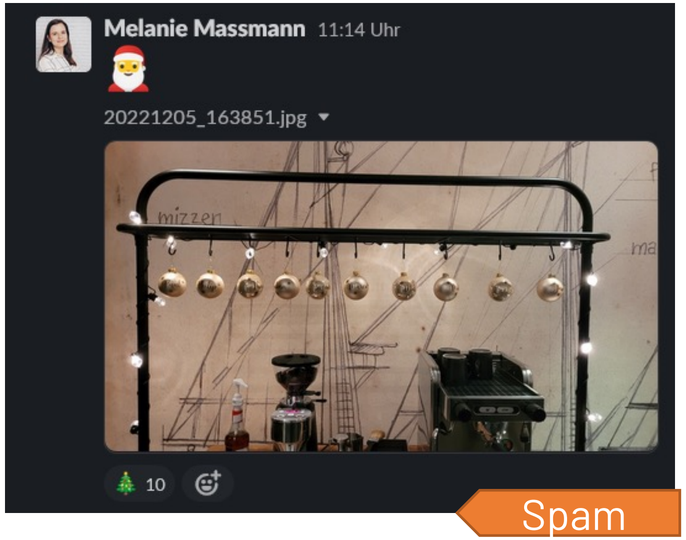
1. Data is transferred from various sources into vector documents using language models
2. Consideration of the data source for semantic information enrichment
3. Using open-source technologies, the questions are answered by a retriever-reader pipeline
4. Via a chatbot, questions can be asked as well as feedback on the answers can be collected
5. User feedback can be used for model optimization

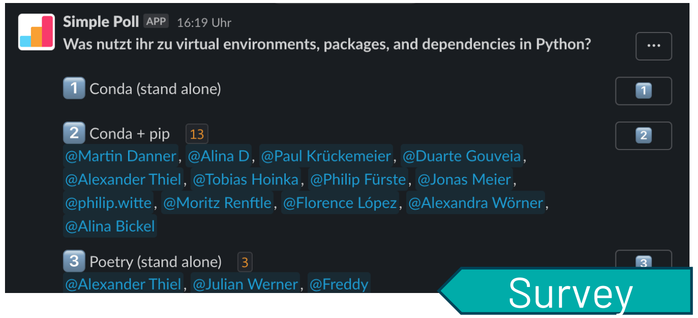
1 The use of heterogeneous data sources requires subject-specific data preparation

scieneers DRIVEN BY DATA

Examples for different data sources







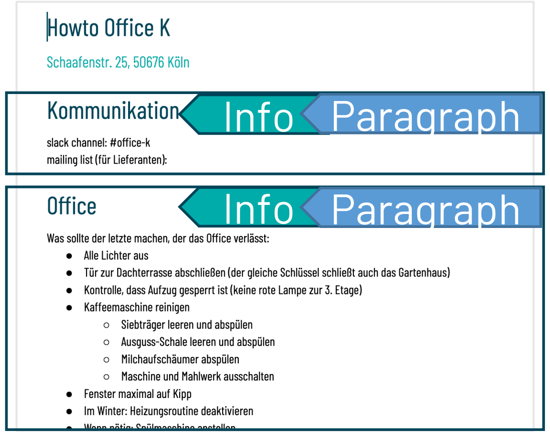
Explanation

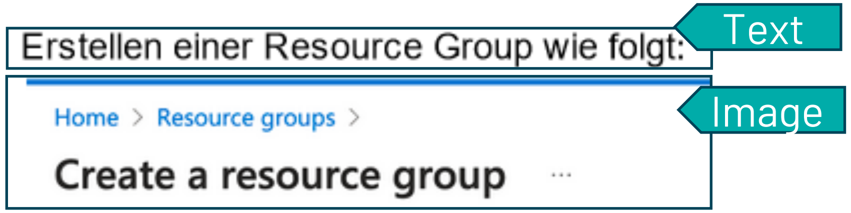
- **Unstructured** data
- Often **many messages** per day
- A lot of meta information
- Sometimes the messages are written carelessly
- **Threads** and possibly **replies**
- Surveys with plugins
- **Reactions** to messages
- Many short messages or emojis and images
- Messages with **no relevant context**
- Partly only **short-lived information**

1

The use of heterogeneous data sources requires subject-specific data preparation

Examples for different data sources





Explanation

- **Structured data**
- Less meta information
- Images between individual text blocks
- **Different text lengths**
- Long G Drive documents are **divided into subdocuments** using the heading structure.
- Sometimes **outdated information**
- Special formats such as video recordings or PowerPoint presentations require **special processing methods**.

How do I read the data from Slack and G Drive?

2

Different data sources require different methods of extraction

Slack

- Messages are **read by the bot** from Slack
- Using the **Slack API**
- Replies must be read out separately
- **Rate Limit Handler** necessary so that the application does not crash

G Drive

- **Search** documents **by file name** or **load** all documents **from** a **folder**
- Using the **Drive API** from Google
- Different document types (Google Documents vs. Microsoft Documents) require **different download methods**

Handling of heterogeneous data sources with individual data ingest pipelines is very important!

Creating additional document metadata

Labelling noisy documents with little information as spam

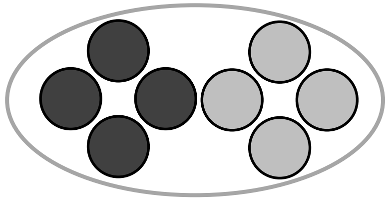
- SetFit: an **efficient** and **prompt-free framework** for **few-shot fine-tuning** of sentence transformers
- Achieves high accuracy with small training datasets

https://github.com/huggingface/setfit

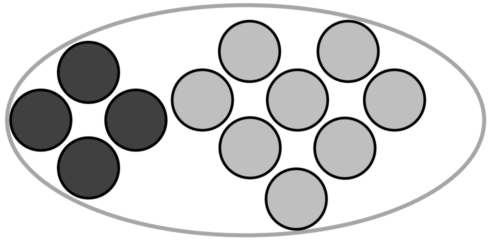
Creating additional document metadata

Labelling noisy documents with little information as spam

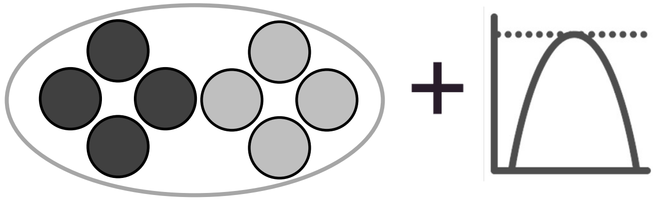
default

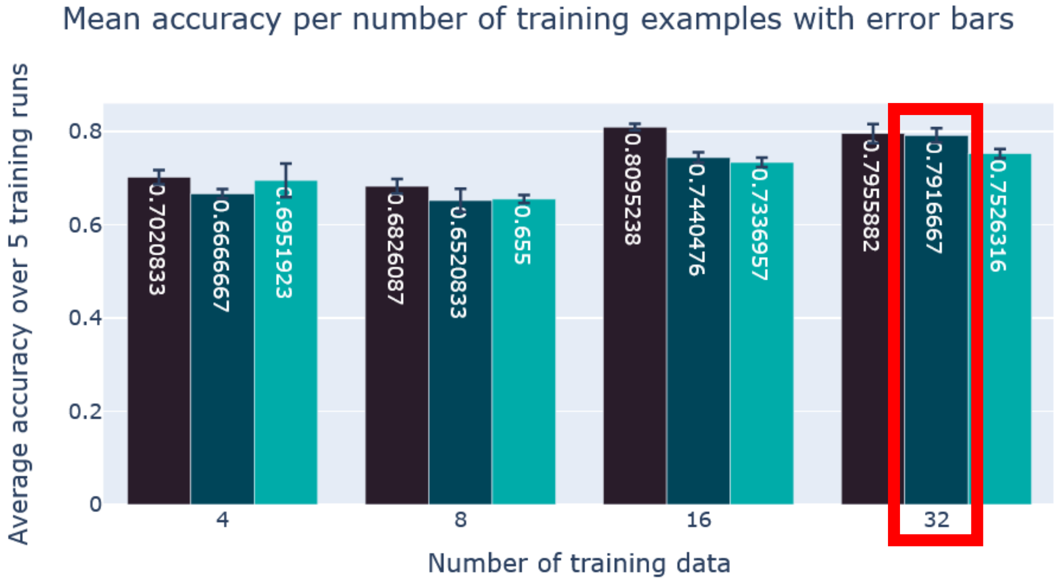


more training data „not spam“



balanced dataset + threshold

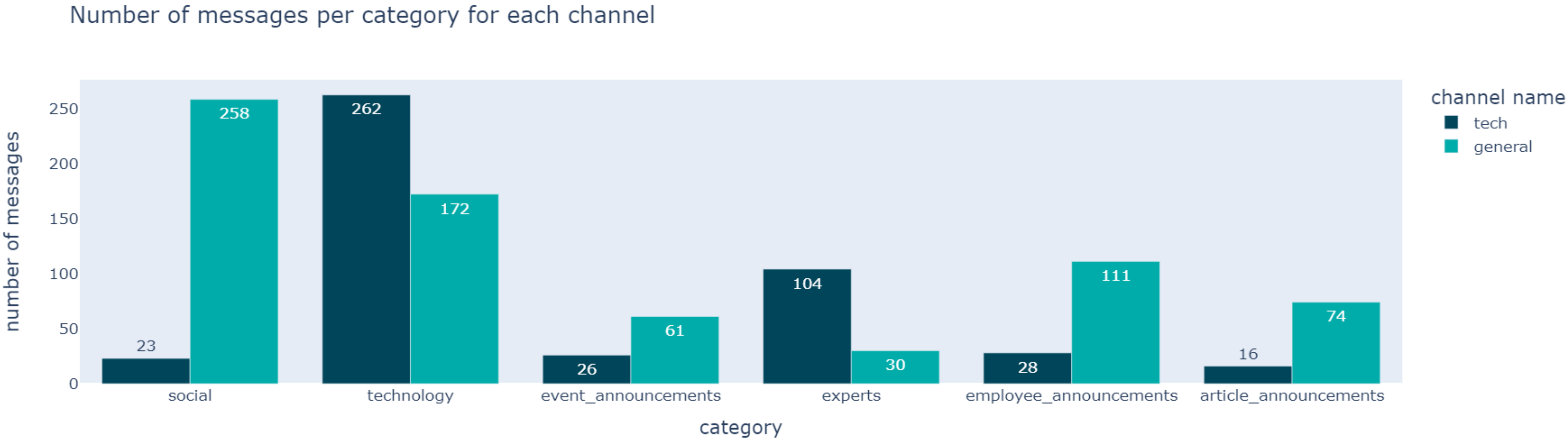




Creating additional document metadata

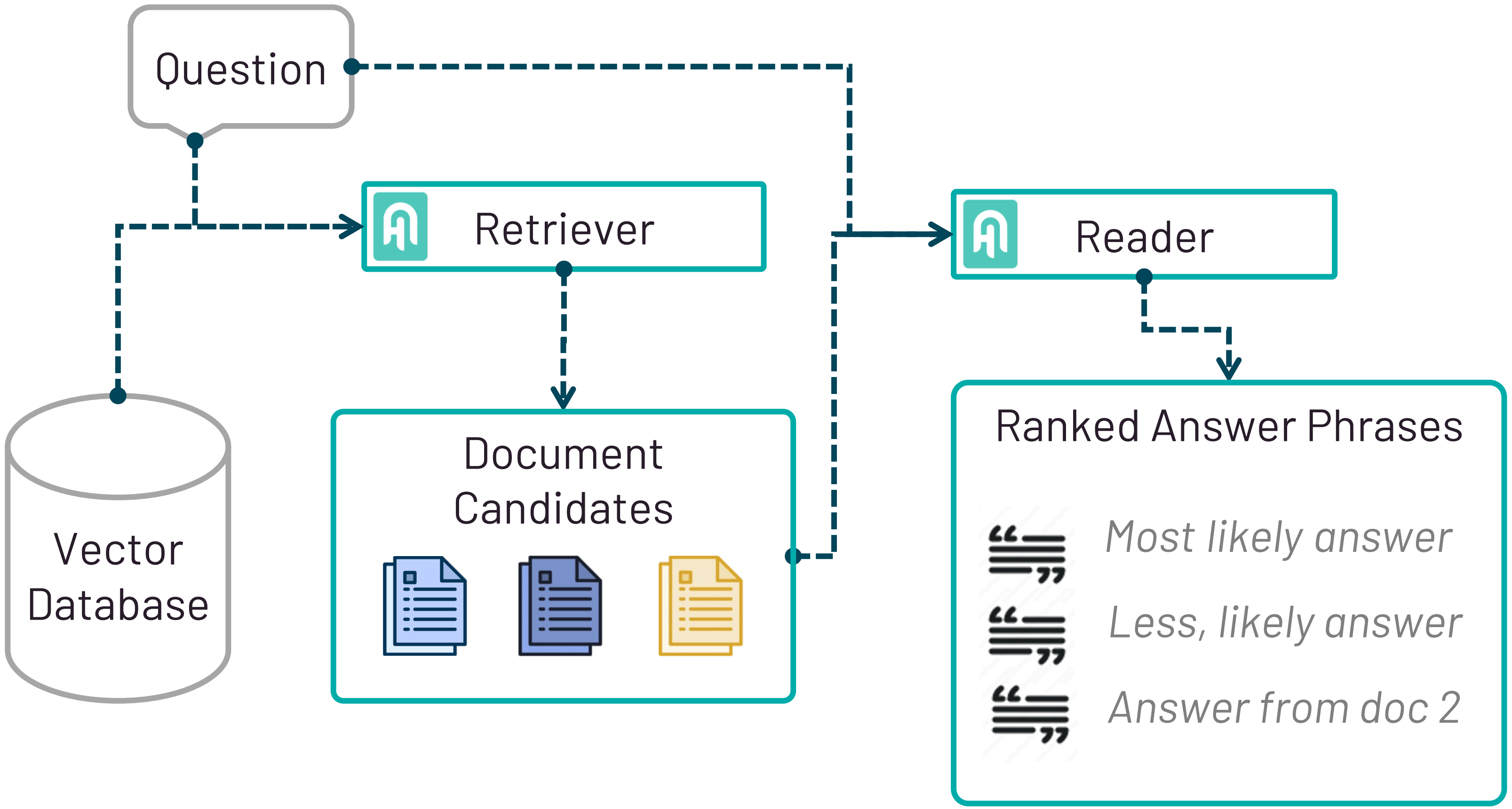
Giving documents some meaning helps to understand the model better

- Slack messages can be assigned to different categories, such as expert questions or event announcements.
- Use a SetFit model to classify messages



3 The retriever-reader Concept

Logical representation of a reader-retriever pipeline in Haystack



Components of the pipeline

Retriever:

- Simple algorithm for identifying relevant candidates
- Evaluation of the relevance of the documents, e.g., by means of TF-IDF, BM25 or EmbeddingRetriever

Reader:

- Extract potential answer phrases from the documents
- Creates a ranking using deep learning-based relevance scoring

Retriever: Finding relevant documents

term-based and vector-based retrieval methods have different strength and pitfalls

- **Term-based** (sparse retriever): based on counting the occurrences of words resulting in very sparse vectors with length = vocab size
  - **Pros**: simple, fast, well explainable
  - **Cons**: relies on exact keyword matches between query and text
    - **Example**: "Who are the developers of ScieneerQA?" vs. "Who built the question-answering chatbot?"
- **Vector-based** (dense retriever): use neural network models to create „dense" embedding vectors
  - **Pros**: captures semantic similarity
  - **Cons**: more computationally intensive use, initial training of the model

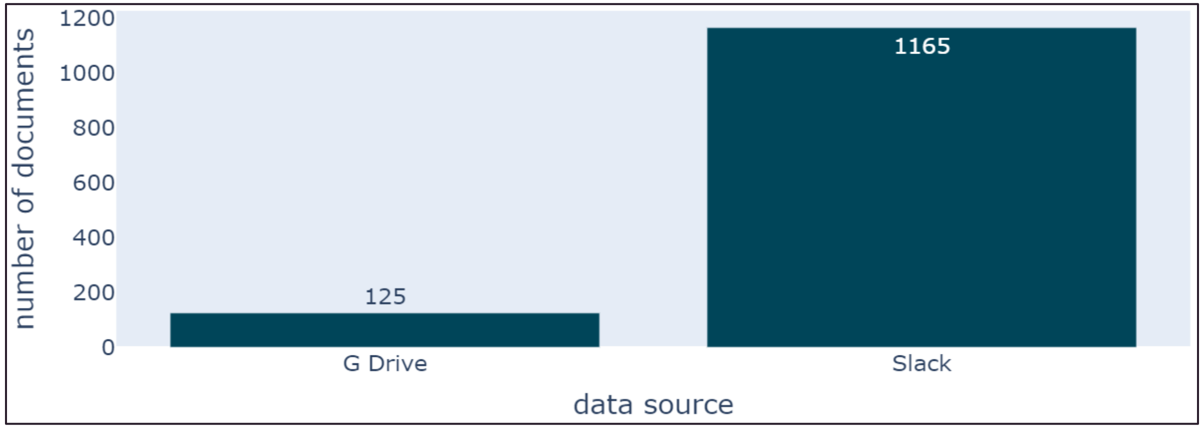
- Term-based retrievers are a good start, but vector-based retrievers often perform better in the real world
- Be careful when evaluating retrievers: The evaluation dataset must reflect real user questions!

https://haystack.deepset.ai/tutorials/06_better_retrieval_via_embedding_retrieval

Retriever: Finding relevant documents

Hybrid-Search combines the best out of both worlds

- Various experiments and test procedures showed us that a **combination of both worked best for us**
- Advantage of using multiple retrievers: **multiple perspectives**
- We currently have three retrievers: BM25, vector-based retriever for G Drive documents, vector-based retriever for G Drive and Slack
- Use a pipeline where the results from all three retrievers are joined and the best 10 are given to the reader
- The number of documents that are put into the reader can be changed
  - Has an impact on the performance of the model



Reader: Create an answer

Extractive and abstractive reader models require totally different paradimes

- Two common types of reader:
  - **Extractive**: extract the answer from the given context
    - **Pros**: allows to determine exactly the source from which the answer comes, labelling is easy
    - **Cons**: an answer cannot always be extracted from exactly one paragraph
  - **Abstractive**: generate an answer from the context that correctly answers the question
    - **Pros**: can create rich and more accurate answers
    - **Cons**: difficult to create suitable labels, difficult to measure the similarity between label and prediction

- We currently only use an extractive reader
- We plan that ChatGPT will later provide the abstractive power to combine extracted information

https://huggingface.co/docs/transformers/tasks/question_answering

Streamlit dashboard for monitoring the ML model

Evaluation dashboard with test use cases is critical to identifying problem cases

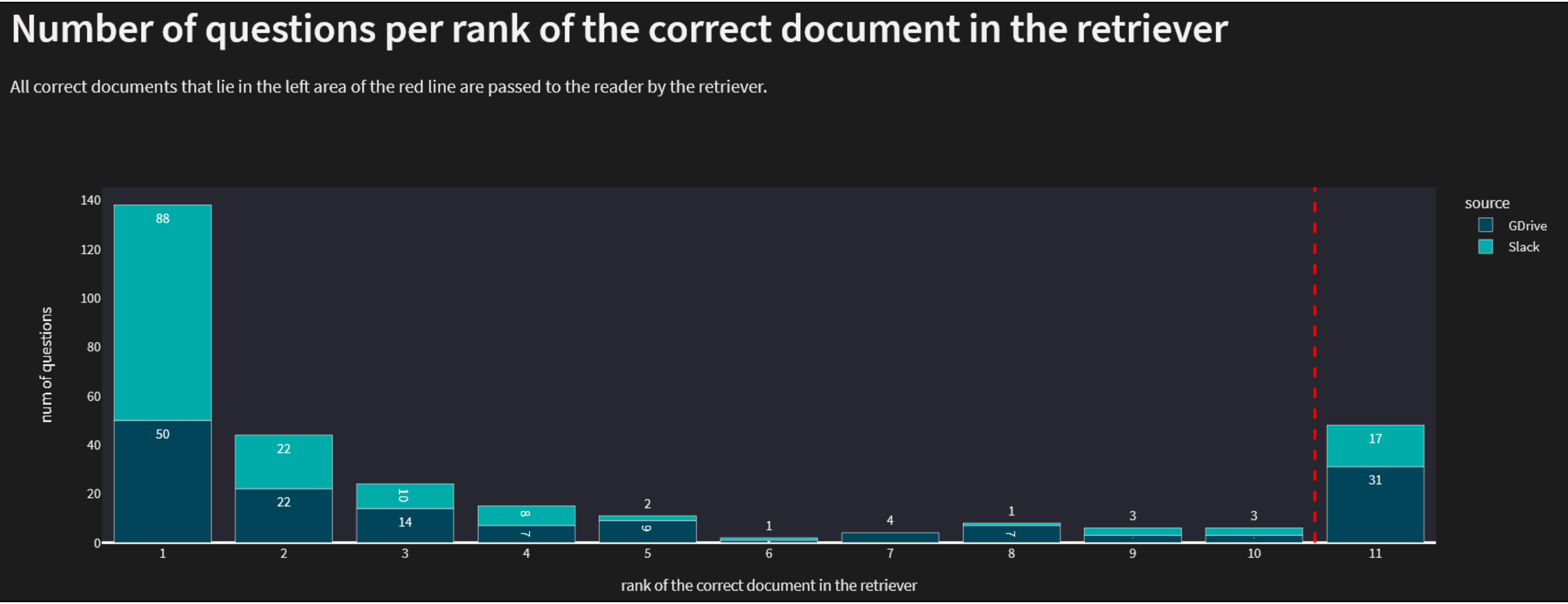
- Streamlit: transforms Python scripts into interactive web applications
- **Similarity** between prediction and target is **measured by** the **ROUGE-F1 score**
  - ROUGE: a Set of metrics for evaluating automatic summarization of texts as well as machine translations
- Some measures on the number of correct and incorrect answers
- **Tracking** the **position** of the document with the correct answer in the retriever
- How well do retriever and reader perform?

- Challenge: What are good/bad labels? The choice of questions influence the metrics!

https://www.freecodecamp.org/news/what-is-rouge-and-how-it-works-for-evaluation-of-summaries-e059fb8ac840/

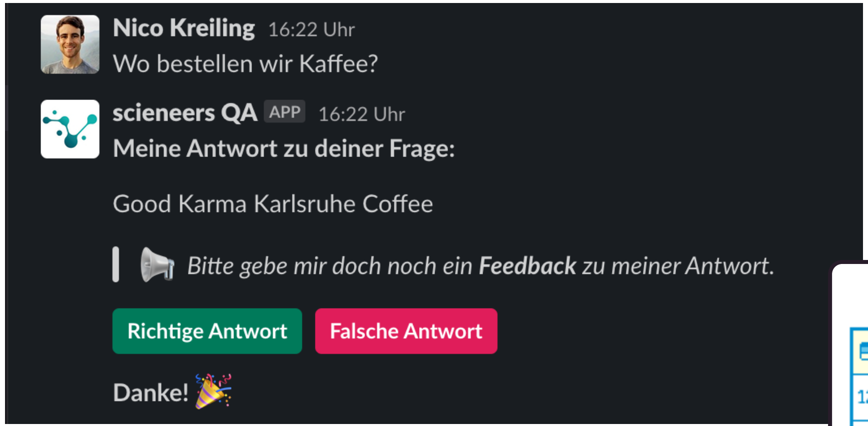
Streamlit dashboard for monitoring the ML model

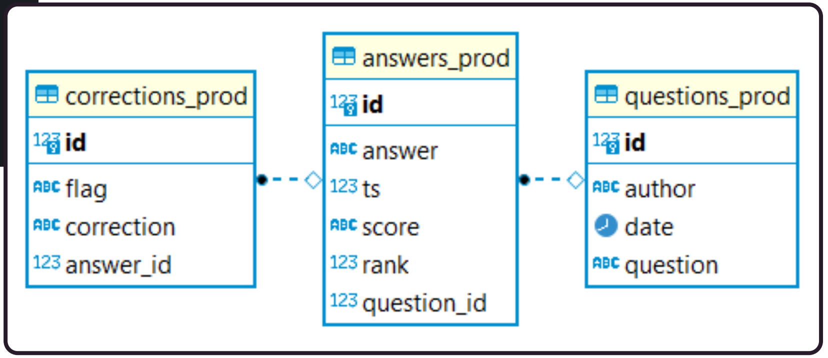
Evaluation dashboard with test use cases is critical to identifying problem cases



4 A chatbot serves as a natural interface for bi-directional user communication

Sample communication flow with the bot





id	author	date	question	answer	score	rank	flag
1	Alina Bickel	2022-11-23 08:17:36.132	Wo bestellen wir Kaffee?	Good Karma Karlsruhe Coffee	0.9564864635467529	1	Correct

Explanation

User can ask the bot questions in both private chats and channels.

Bot answers the question with a simple API call to the QA pipeline.

User can **confirm** or **correct** the answer via buttons in the chat.

The bot stores this information in a database, which can be used for **quality assessment** and **further development**.

Chatbot Design

How we created the chatbot

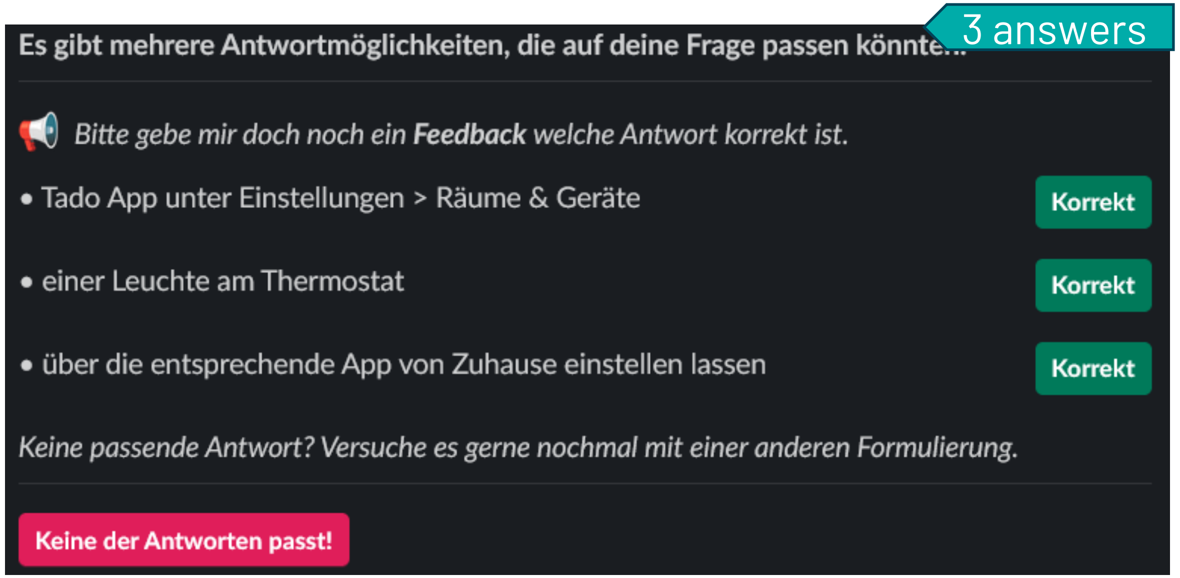
- Use of the framework „Bolt“
  - Quick start in programming
- Welcome message when opening a direct message with the bot
- App interactions and events currently run over a WebSocket connection
  - Using the SocketModeHandler from Bolt
  - No public, static HTTP endpoint necessary

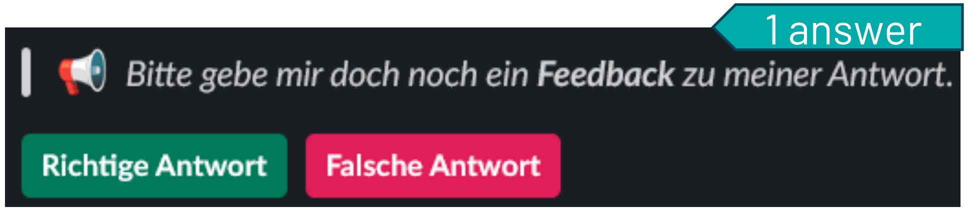
https://slack.dev/bolt-python/concepts

https://api.slack.com/apis/connections/socket

5 How we collect information back

Example of how the buttons are displayed





Explanation

Users can give feedback on the answers **via buttons**.

If the answer is wrong, the **correct answer** can be written **as a reply**.

Using the **object relational mapping** technique for reading the data.

Mapping of correction to the answer is made via timestamp of the answer from the bot.

What are the most important insights?

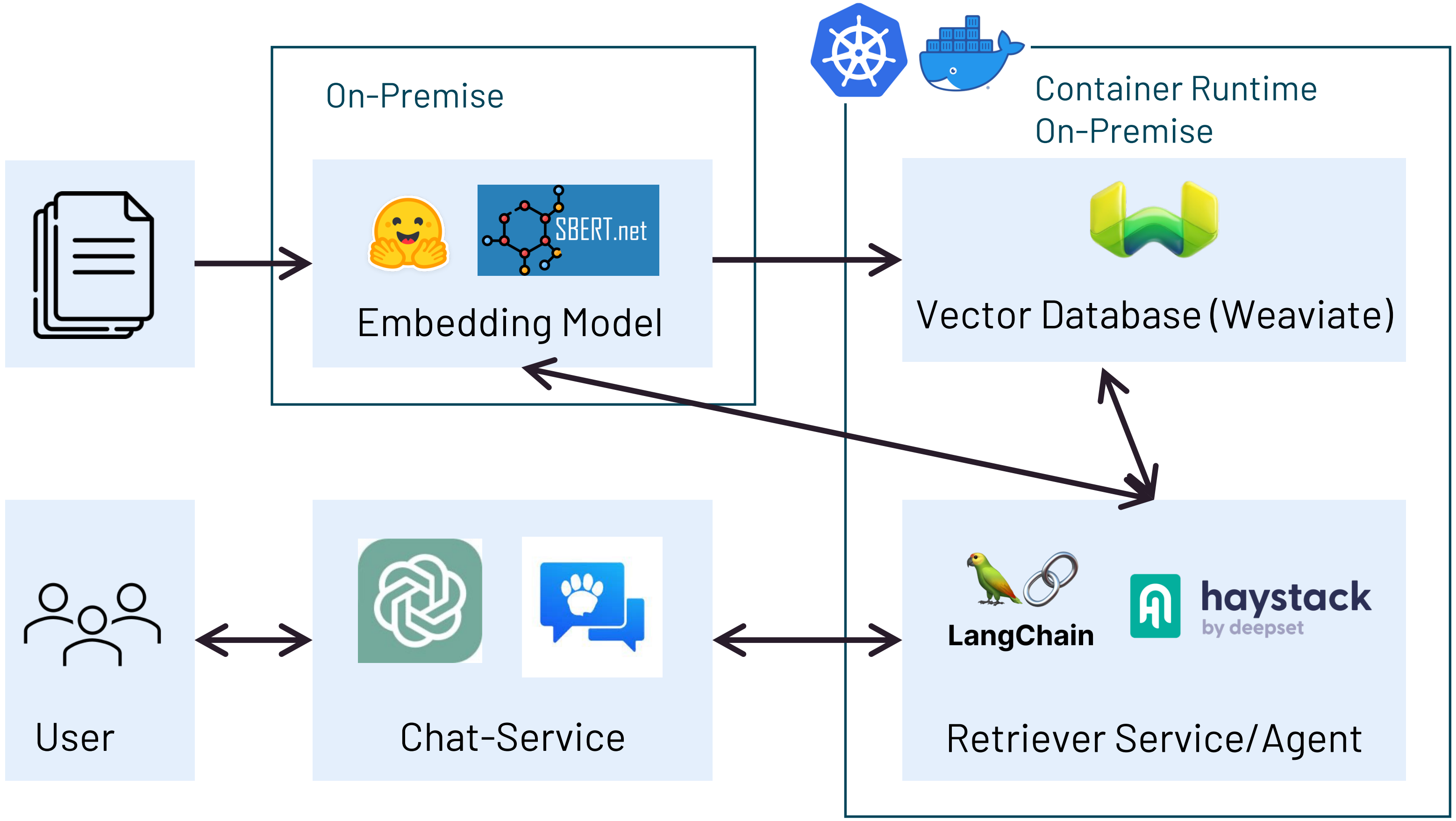
Analysis and adaptation of existing systems to your own data is crucial

- Haystack and Weaviate work well in principle
- Handling the heterogeneous data with specific pre-processing pipelines is important
- Very **different document lengths** are problematic
- The G Drive documents should be well structured
- Filtering and down-sampling of the high-quantity but verbose Slack messages is important
- A good **evaluation dashboard** with test use cases is critical to identifying problem cases
- Don’t fool yourself with non-representative evaluation datasets

LLMs are the Future (also of our ScieneersQA)

Extended by some retrieval agent to access internal knowledge

How the next version architecture diagram will look like



Workflow description

- Still pre-processing and embedding all internal documents to store them in a vector database, so that they can get retrieved using a **retriever module** or a **full QA-pipeline.**
- Using some closed-source LLM endpoint or hosting an open LLM to act as **user chat agent**.
- Enable the LLM to lookup our data using some phrase. For example, by adding a **ChatGPT Plugin** or a **Haystack Agent**.

Q&A