

Marcel: A Lightweight and Open-Source Conversational Agent for University Student Support

Jan Trienes Anastasiia Derzhanskaia Roland Schwarzkopf
Markus Mühling Jörg Schlötterer Christin Seifert

Marburg University

{jan.trienes, christin.seifert}@uni-marburg.de

Abstract

We present *Marcel*, a lightweight and open-source conversational agent designed to support prospective students with admission-related inquiries. The system aims to provide fast and personalized responses, while reducing workload of university staff. We employ retrieval-augmented generation to ground answers in university resources and to provide users with verifiable, contextually relevant information. We introduce a Frequently Asked Question (FAQ) retriever that maps user questions to knowledge-base entries, which allows administrators to steer retrieval, and improves over standard dense/hybrid retrieval strategies. The system is engineered for easy deployment in resource-constrained academic settings. We detail the system architecture, provide a technical evaluation of its components, and report insights from a real-world deployment.

 github.com/aix-group/marcel-chat

 youtu.be/uLCB2R6szz4

1 Introduction

Prospective university students often face challenges navigating the admission requirements. While many universities are investing in student support services, these typically do not grow at the same rate as many study programs. To contextualize this problem in a real-world example, we consider the M.Sc Data Science program at our institution. The growing interest in this subject and a recent internationalization of the program lead to a stark increase in applications. However, many applications had to be rejected because formal criteria or prerequisites were not met. This substantially increased the workload of university staff and led to missed opportunities for students. Expanding support services with alternative information channels appears necessary to better inform applicants and to reduce the workload of university staff.

A potential solution to this problem are conversational search systems like chatbots (Zamani et al.,

2023; Mo et al., 2025). We posit that these systems make it easier for students to find and understand relevant information due to their interactivity. Recent advances in Retrieval Augmented Generation (RAG, Lewis et al., 2020) allow language models to provide grounded and verifiable answers based on university resources. However, operating such a RAG-based system in a university setting is challenging. Many universities require on-premise deployments to comply with privacy laws, have limited computational resources, and need cost-effective systems that are easy to customize, maintain and monitor. Several universities are actively developing or using such chatbots (Nguyen and Quan, 2025; Odede and Frommholz, 2024; Cherumanal et al., 2024; Brereton, 2021) which indicates high demand. However, they are typically closed-source which limits others to build on those efforts.

To bridge this gap, we introduce *Marcel* a student chatbot aimed at answering enrollment-related questions.¹ This demo makes three contributions:

(1) **FAQ Retriever** (Section 2). We introduce a novel retriever which maps user queries to a set of curated frequently asked questions with associated answer-relevant documents. This component allows administrators to steer retrieval, and improves over dense/hybrid retrieval strategies.

(2) **Efficient Deployment** (Section 3). We outline our design decisions to facilitate deployments with minimal infrastructure (i.e., a single node with a GPU). We empirically evaluate several retrieval and generation strategies to better understand quality-efficiency tradeoffs in this task.

(3) **Practical Insights** (Section 4). We share qualitative insights from deploying the chatbot in practice. To support adoption, we release our system as containerized software along with support-

¹For this demo, we focus on enrollment-related questions for the M.Sc Data Science program at our institution. However, all design decisions are made to generalize to other information-seeking tasks in the university setting.

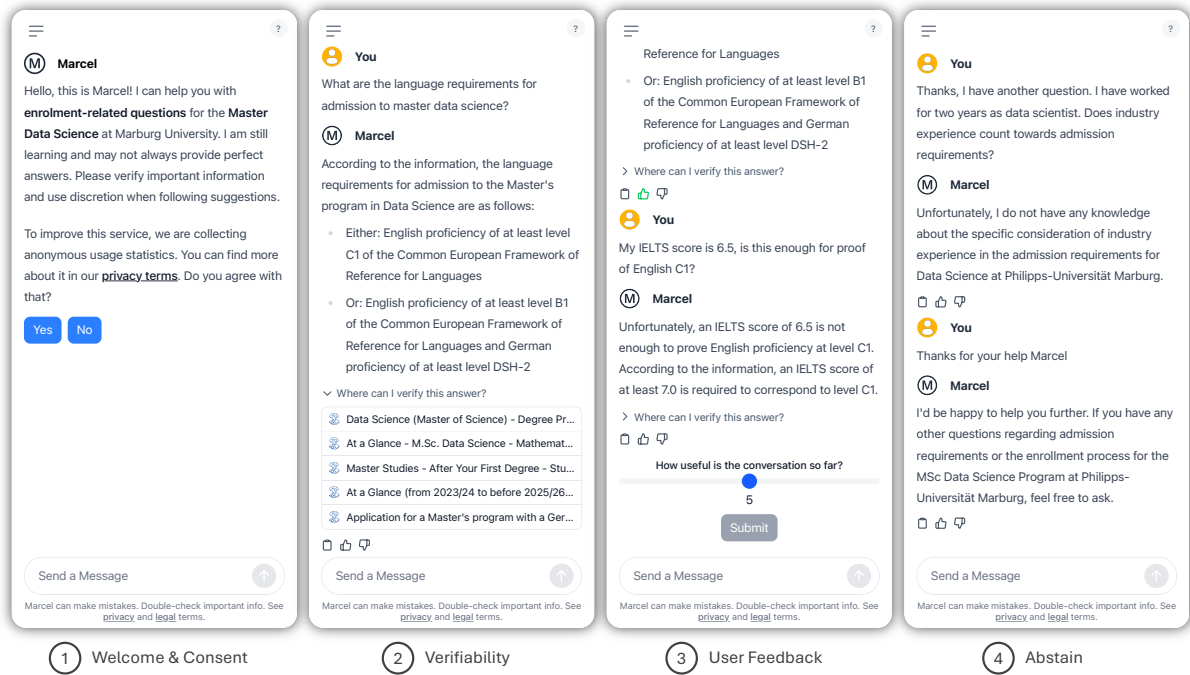


Figure 1: Overview of student-facing UI. ① Welcome and consent are shown on first use. ② For each generated answer, Marcel provides the list of source documents which are linked to the respective websites. ③ Users can rate individual responses using a thumbs up/down button. The overall conversation can be rated after three interactions on a 10-point Likert scale. ④ Marcel abstains from answering questions that are not reflected in the underlying knowledge base. In those cases, no list of source documents is shown. An admin-facing UI is given in Figure 6.

ing artifacts such as content scraper and deployment scripts to enable customization.²

2 System Description

2.1 Question Answering

Document retrieval. *Marcel* uses RAG to provide answers grounded in university resources. To improve retrieval precision and add a layer of administrative control, we introduce an FAQ retriever that maps queries to a set of curated questions, each linked to one or more relevant documents. The closest FAQ is selected through semantic similarity of question embeddings. Intuitively, FAQ retrieval acts as a lightweight intent classifier. We combine it with standard retrievers (e.g., BM25) to provide fallbacks for question intents falling outside the FAQ. Although FAQ curation requires manual effort, many universities maintain FAQ-like resources (e.g., for student support services) that can be bootstrapped. For our deployment, we worked with university staff to establish 36 FAQs, reflecting most enrollment related information-needs. We provide example FAQs in Appendix, Table 5.

Answer generation. An instruction-tuned Large

Language Model (LLM) generates answers conditioned on retrieved documents. To improve answer grounding and dialogue handling, answering is done in two stages. First, an LLM-based classifier determines whether a query requires retrieval or is chit-chat or a meta-question (e.g., “What topics can you help with?”). If so, the system responds without retrieval. Otherwise, the LLM is instructed to answer only if the retrieved documents provide sufficient information, or to abstain and return a fallback response (e.g., “Unfortunately, I cannot answer based on...”). See Appendix B for the prompts of query classifier and answer generator.

Knowledge base. We scrape enrollment-related documents from university websites (e.g., study descriptions, admission requirements and procedures, course catalog, and accommodation services). Curating the knowledge base through scraping is efficient, since latest information is published on the websites. Periodically re-running the scraper captures changes automatically (e.g., updated admission requirements). Pre-processing of scraped content is kept minimal: to reduce token count in downstream generation, we convert HTML to Markdown and replace links with numeric IDs. We avoid splitting documents to preserve context. The

²All software is released under the MIT license.

corpus consists of 249 documents, with an average length of 712 tokens using the NLTK tokenizer.

2.2 User Interaction

We support two user groups: (i) students seeking information, and (ii) administrators wanting to monitor and evaluate the system. The student-facing UI adopts familiar conventions of chat interfaces and is fully responsive (Figure 1). On first visit, a consent dialog informs users about data collection and chatbot limitations. The disclaimer states that generated answers may be inaccurate and must be verified. Users need to confirm that they have read this message before continuing. We discuss broader ethical implications of the chatbot at the end of the paper.

Responses are shown alongside links to retrieved sources to support verification. Users can rate each response with a thumbs-up/down, and after several turns, provide conversation-level feedback on a Likert scale. If a query is not answerable with retrieved context, the system abstains and hides sources. The admin UI shows usage statistics over a user-defined period and allows to review full conversations and ratings (Appendix, Figure 6).

2.3 Software Architecture and Deployment

Marcel is designed for fast deployment in resource-constrained academic settings. The chatbot is implemented as a containerized architecture with four components: a JavaScript frontend (Vue.JS), a Python backend (FastAPI), a relational database (MariaDB), and an LLM inference engine (vLLM, Kwon et al., 2023). A reverse proxy (nginx) routes traffic and handles HTTPs encryption. We run Marcel on a node with 2 A100 GPUs for vLLM³ and a VM (4 cores, 4GB RAM) for the remaining containers. We implement best practices for container security by OWASP (2025), including (i) container isolation, (ii) minimal rights, (iii) not exposing the Docker socket. Automated, reproducible deployments and updates are done using Ansible. To support adoption by other institutions, we share both the Ansible scripts and regular docker-compose files as a blueprint.

3 Technical Evaluation

We conduct a technical evaluation of *Marcel’s* core components to better understand the quality and

efficiency trade-offs in this resource-constrained university chatbot setting. Our experiments are guided by the following research questions:

- RQ1.** How does the FAQ-augmented retriever compare to standard retrieval strategies?
- RQ2.** To what extent can a second-stage reranker improve retrieval quality?
- RQ3.** How well do small-scale, open-weights generators perform in this domain?
- RQ4.** What is the impact of quantization on generation quality and efficiency?

3.1 Experimental Systems

Retrievers. To address RQ1 and RQ2, we evaluate diverse retrieval strategies commonly used in production systems, spanning lexical, dense, and hybrid methods, with and without reranking.

- **BM25** (Robertson and Zaragoza, 2009). Standard lexical search, which is a strong baseline in practical settings.
- **Dense.** Document relevance is determined by the similarity between query and document embeddings. Document embeddings are precomputed during indexing using models optimized for semantic search in sentence-transformers (Reimers and Gurevych, 2019). Specifically: *MiniLM* (22.7M params) and *MS MARCO Bert* (109M params).
- **HyDE** (Gao et al., 2023). An LLM is prompted to generate n artificial documents that could plausibly answer a given query. The average embedding of these documents is used as the query, similar to dense retrieval. To understand how generator scale affects HyDE performance, we test the Gemma 3 model family (Kamath et al., 2025). We set $n = 3$ and use the same embedding model as the dense retriever (MS MARCO).
- **FAQ.** Retrieving relevant documents via FAQ matching (see Section 2.1).

We evaluate combinations of above retrievers by merging document rankings through reciprocal rank fusion (Cormack et al., 2009). To complement these retrievers, we evaluate second-stage reranking using cross-encoders (**Rerank**).⁴ For retrieval evaluation, we take the top $k = 50$ documents. For

³The newer Gemma 27b model runs on a single A100. With 4bit quantization a 3090 is supposed to be sufficient.

⁴We use *mxbai-rerank-base-v1*, which performs well on general benchmarks (Shakir et al., 2024); See Appendix A for an evaluation of other representative models in the 33M to 435M parameter range.

generation, we restrict to the top $k = 5$ documents to meet latency and context length constraints.

Generators. We seek a generator that is (i) open-weight, allowing for a local and privacy-compliant operation, and (ii) that is deployable in hardware constrained settings (e.g., 1–2 GPUs). The current deployment of *Marcel* uses Llama 3.1 (Grattafiori et al., 2024), in the 70B variant with int8 quantization. For comparison, we benchmark a second candidate, Gemma 3 (Kamath et al., 2025), because of its strong performance in the Chatbot Arena (Chiang et al., 2024).⁵ To address RQ3 and RQ4, we evaluate all available model sizes for both Llama and Gemma families, and assess the impact of int8 quantization for Llama models. As retriever we use BM25+FAQ. Additionally, we use an Oracle to estimate generation quality under optimal retrieval. For each query, we generate three outputs with temperature sampling ($\tau = 0.7$), and report evaluation metrics averaged over these generations. The system prompt is provided in Appendix, Listing 1.

3.2 Evaluation Protocol

Data. In the initial development phase of *Marcel*, we lacked real user queries directed at the chatbot. Therefore, we bootstrapped an evaluation dataset from student emails sent to university support offices. This provides a realistic approximation of the information needs that end-users have. To simplify the analysis, we focus on single-turn interactions.

Emails were collected and anonymized during a time frame of 6 months. We performed light deduplication of overrepresented information needs. Each question is associated with a manually written ground-truth response and the relevant sources in the knowledge base (i.e., document IDs). We distinguish between questions that are *answerable* and *unanswerable* with the knowledge base, to assess the generator’s ability to abstain when the retrieved context does not provide relevant information (e.g., questions about visa issues, late arrivals, conditional accepts, etc.). See Table 1 for an overview of the dataset and Appendix, Table 6 for examples.

Metrics. We evaluate retrieval by Mean Reciprocal Rank (**MRR**) and Recall at Cutoff (**R@K**). To evaluate generation, we employ both reference-based and reference-free metrics, using LLM-as-

⁵At the time of study (June 2025), Gemma 3 was the highest ranked model that can be deployed on a single GPU (ranked 28/254). The next better open-weights model was Qwen 3 with 235B parameters (ranked 23/254).

Statistic	All	Answerable	Unanswerable
Questions	95	76	19
Question length	43.3	40.5	54.3
Answer length	45.5	46.0	43.6
Answer sources	1.0	1.3	0.0
FAQs	36	25	11
FAQ length	9.7	8.9	12.8

Table 1: Summary statistics of the evaluation dataset.

a-Judge. For reference-based metrics, we report **ROUGE-1** (Lin, 2004) and **BERTScore** (Zhang et al., 2020). For reference-free evaluation, we report **Answer Faithfulness** and **Answer Relevance** (Es et al., 2024). Answer faithfulness estimates the fraction of answer claims supported by the retrieved context. Answer relevance generates n hypothetical questions from the answer, and measures their similarity to the original query.

Additionally, we evaluate the generator’s ability to abstain when a query cannot be answered with the knowledge base. Abstaining is important, as any answer generated from the parametric knowledge of the LLM is not grounded in the retrieved context and may therefore be incorrect, irrelevant or outdated. To this end, we define **Selectivity**. Let $a_{i,j} \in \{0, 1\}$ indicate whether generation j is an attempt at answering query i , and let $y_i \in \{0, 1\}$ denote the ground-truth obtained from manual annotation. Given N queries and M generations,

$$\text{Selectivity} = \frac{1}{N} \sum_{i=1}^N \left(\frac{1}{M} \sum_{j=1}^M \mathbb{1}[a_{i,j} = y_i] \right),$$

where $\mathbb{1}$ is the indicator function. Selectivity ranges from 0 to 1, where higher is better. To obtain $a_{i,j}$, we instruct an LLM to detect hedging phrases (e.g., *I don’t know*, *The documents do not provide any information*). See prompt in Appendix, Listing 4. We report all generator metrics separately for *answerable* and *unanswerable* queries. This allows us to understand if the system abstains correctly when questions cannot be answered.

Finally, we report computational efficiency as **seconds per query**. Experiments were conducted on a Slurm node with four CPU cores and two A100 GPUs (80GB), with timings collected after pipeline warmup. All LLM-as-a-Judge metrics are computed using Llama 3.1 70B (int8 quantized).⁶

⁶While Llama could be favoring its own generations, Panickssery et al. (2024) showed that this mainly applies to pairwise evaluations, not direct ratings used in our experiments.

Retriever	MRR	R@1	R@5	R@50	sec/q	sec/q [†]
<i>Baseline</i>						
BM25	0.26	0.09	0.37	0.74	0.02	—
+ Dense (MINILM)	0.31	0.15	0.44	0.82	0.06	0.04
+ Dense (MARCO)	0.40	0.21	0.50	0.78	0.23	0.06
<i>Hyde (Gao et al., 2023)</i>						
+ HyDE (G-1B)	0.28	0.13	0.34	0.73	—	4.81
+ HyDE (G-4B)	0.38	0.25	0.41	0.82	—	5.22
+ HyDE (G-27B)	0.44	0.29	0.49	0.84	—	20.92
<i>FAQ Retriever (Ours)</i>						
+ FAQ (MINILM)	0.70*	0.57*	0.77*	0.84	0.05	0.03
+ FAQ (MARCO)	0.61*	0.47*	0.63	0.80	0.19	0.03
<i>Reranker (mxbai-rerank-base-v1)</i>						
+ Dense + Rerank	0.46	0.33	0.48	0.78	—	0.59
+ HyDE + Rerank	0.53	0.38	0.55	0.83	—	21.41
+ FAQ + Rerank	0.48	0.34	0.49	0.84	—	0.56

Table 2: Evaluation of retrieval strategies. Statistically significant differences relative to baseline (BM25 + DENSE MARCO) are denoted by * at a significance level of $\alpha = 0.01/40$ (Bonferroni correction). Seconds per query are reported on CPU (sec/q) and GPU (sec/q[†]), while HyDE and Rerank are only tractable on GPU.

3.3 Results

We present the evaluation of retrievers in Table 2 and the evaluation of generators in Table 3.

Effectiveness of FAQ retriever (RQ1). The FAQ retriever significantly outperforms standard retrieval methods, achieving a 75% increase in MRR over the BM25+Dense baseline. This gain in early-rank retrieval quality is particularly valuable for downstream generation, which typically benefits from smaller context sizes both in output quality and throughput. Importantly, strong performance is achieved even with lightweight embeddings (MiniLM, 22.7M parameters), enabling fast, CPU-friendly retrieval with low latency (0.05 seconds). Overall, these findings suggest that the FAQ retriever is an effective mechanism for administrators to steer retrieval, with the trade-off of manual FAQ curation and maintenance.

Effectiveness of reranking (RQ2). We next examine the impact of second-stage reranking on retrieval quality. While both BM25+Dense and BM25+HyDE retrieval show modest improvements at early ranks, the quality of FAQ retrieval is substantially degraded ($\approx 31\%$ drop in MRR). This suggests that current rerankers may not generalize to this domain. One potential mitigation strategy is to exclude FAQ-retrieved documents from reranking. However, this approach is only feasible when

the FAQ retriever reliably returns relevant documents. Future work could investigate techniques to dynamically select the most appropriate retriever per query, or to fine-tune rerankers.

Generator evaluation (RQ3). We find that Gemma (12B, 27B) and Llama (70B) perform comparably, especially on the *answerable* subset of queries (Table 3). Gemma (12B) strikes a good balance between quality and resource efficiency, showing that it may not be necessary to resort to the largest available models. We note an interesting model-dependent trend on response length: larger Llama models tend to produce longer responses, while Gemma models show the opposite pattern. However, we cannot conclude that either behavior leads to higher response quality.

Substituting the FAQ retriever with an oracle retriever results in minimal changes in performance on answerable queries. This highlights the quality of the FAQ retriever, but also indicates that the current evaluation protocol may be insensitive to improvements beyond a certain threshold. On unanswerable queries, the use of an oracle retriever significantly increases the rate of correct abstentions (0.68 $\rightarrow$ 0.93), suggesting that more conservative retrieval can further improve faithfulness. Additionally, it may be worthwhile to investigate alternative strategies for classifying whether or not a question is answerable given the retrieved context (Peng et al., 2025). Currently, this task is left to the answer generation prompt (cf. Section 2.1) which may leave room for improvement.

Impact of quantization (RQ4). We find that int8 quantization has no noticeable effect on generation quality for both Llama 8B and 70B. At the same time, it significantly reduces inference latency (speedup of $\approx 20\%$ – 42%). This makes quantization a suitable technique to reduce hardware requirements for practical deployments of *Marcel*.

4 Insights from End-User Deployment

We deployed Marcel during the final weeks of the current admissions cycle (winter term 2025/26). Despite limited user engagement due to the timing, this deployment demonstrates production-readiness and allowed us to collect qualitative insights from end-user interactions. Over a nine-week period, Marcel served 92 users, exchanging 926 messages in 126 conversations, without any system errors or downtime. Below, we discuss several qualita-

Generator	ROUGE	BERTScore	Faithfulness	Relevance	Selectivity	Average	$ \hat{y} $	sec/q	GPUs
Answerable with knowledge base									
Gemma 3 (1B)	0.17	-0.03	0.33	0.73	0.38	0.32	142	7.2 [†]	1
Gemma 3 (4B)	0.24	0.04	0.56	0.72	0.88	0.49	206	3.3	1
Gemma 3 (12B)	0.30	0.15	0.76	0.71	0.79	0.54	113	4.1	1
Gemma 3 (27B)	0.31	0.16	0.71	0.70	0.88	0.55	114	8.0	1
→ Oracle retriever	0.35	0.18	0.63	0.68	0.91	0.55	126	6.8	1
Llama 3.1 (8B)	0.22	0.10	0.54	0.74	0.44	0.41	89	2.4	1
→ int8 quantized	0.20	0.07	0.54	0.75	0.43	0.40	93	1.9	1
Llama 3.1 (70B)	0.29	0.16	0.70	0.74	0.76	0.53	132	12.5	2
→ int8 quantized	0.29	0.16	0.73	0.73	0.80	0.54	130	8.3	2
Not Answerable with knowledge base									
Gemma 3 (1B)	0.16	0.02	0.31	0.64	0.65	0.32	181	7.1 [†]	1
Gemma 3 (4B)	0.16	0.04	0.54	0.61	0.56	0.49	161	2.9	1
Gemma 3 (12B)	0.21	0.14	0.70	0.56	0.75	0.54	59	2.7	1
Gemma 3 (27B)	0.21	0.12	0.75	0.60	0.68	0.55	80	6.5	1
→ Oracle retriever	0.18	0.13	—	0.67	0.93	0.48	43	2.1	1
Llama 3.1 (8B)	0.15	0.13	0.47	0.55	0.91	0.41	53	1.6	1
→ int8 quantized	0.17	0.14	0.52	0.55	0.89	0.40	41	1.0	1
Llama 3.1 (70B)	0.16	0.11	0.55	0.61	0.67	0.53	64	9.5	2
→ int8 quantized	0.18	0.13	0.55	0.66	0.63	0.54	67	5.5	2

Table 3: Evaluation of generators stratified by whether a query is answerable with the knowledge base. Additionally to quality metrics, we report the length of response in tokens ($|\hat{y}|$) and processing time per query (sec/q). [†] = We were unable to obtain reliable timings for Gemma 3 1B due to a vLLM bug (v0.9.1, [vllm/#19575](#)).

tive observations and examples that reflect current limitations of the chatbot.

(1) *Answers are tailored, but not proactive* (Appendix, Figure 2). Some information-needs could be addressed more efficiently through proactivity. For example, instead of answering “Since only 50% your BSc courses are related to Data Science, you may not be admitted,” the chatbot could prompt the user to provide specific course contents, to subsequently match them against admission requirements. However, these interactions require strong model reasoning and carry a high risk of ungrounded inference.

(2) *Knowledge gaps* (Appendix, Figure 3). We encountered several questions that the chatbot abstained from answering. This is caused either by an incomplete knowledge base, or because the website lacks this information. While frequent abstention limits the chatbot usefulness, it improves grounding, and has the beneficial side-effect of identifying ways to improve clarity and completeness of university websites.

(3) *Contextualized queries* (Appendix, Figure 4). The chatbot struggles with queries contextualized in multi-turn dialog. For example, follow-up questions often lack information of previous queries, causing the retriever to surface irrelevant documents. An adjacent problem are underspecified

queries. For example, many user questions do not explicitly call out that the query is about the data science program (e.g., “When is the deadline for application?”), which is a consequence of users entering the chatbot through program-specific pages. Practically, this degrades retrieval quality and can lead to unfocused generations when the retrieved context provides multiple valid or even conflicting answers. A potential solution could be query expansion via decontextualization (Choi et al., 2021).

(4) *Link-seeking queries* (Appendix, Figure 5). While source links are shown alongside answers, the chatbot cannot embed them within responses as they are usually absent from source documents. To enable link-seeking interactions, it may be necessary to embed links in the documents and to instruct the generator to cite them.

(5) *Little uptake of feedback mechanisms*. We observed little user engagement with the feedback mechanisms. The message-level rating (thumbs-up/down) was used only once, and only 11 conversations received a Likert-scale rating. This highlights a general challenge in obtaining direct user feedback. Downstream evaluation may therefore need to focus on answer quality. In addition, latent usage signals such whether users ask followup questions or whether they return to the chatbot could be taken into account.

5 Related Work

RAG (Lewis et al., 2020) is becoming a widely adopted strategy across many domains in conversational search (Zamani et al., 2023). In the educational domain, Nguyen et al. (2021) deployed a chatbot to support admissions using the dialog management framework Rasa (Bocklisch et al., 2017). Significant engagement demonstrates high student demand. Odede and Frommholz (2024) conducted a user study comparing live-chat with university staff to a RAG-chatbot. Participants valued fast responses and low-entry barriers of the chatbot.

Cherumanal et al. (2024) find that intent-based retrieval can be an effective alternative to document retrieval, which supports the overall idea of our FAQ retriever. Closely related, Nguyen and Quan (2025) proposed a two-tiered strategy: if a query is sufficiently similar to a pre-defined FAQ, the system returns a pre-defined answer; otherwise, it falls back to document retrieval. Our system differs in two ways. First, instead of pre-defining answers, we link FAQs to knowledge base documents to ensure grounded and up-to-date answers. Second, instead of imposing a minimum similarity between queries and FAQ entries, we use the similarity as a soft ranking signal combined with other retriever scores which avoids an additional hyperparameter.

In the general domain, document assistants such as TruthReader (Li et al., 2024) and Verba (Weavitate, 2025) follow a similar RAG architecture as our system, but assume user-provided documents. The RAG-framework Onyx (2025) provides rich utilities to connect to heterogeneous data sources, but allows little customization of the retrieval strategy.

Despite these advances, there remains a gap in open-source chatbots designed for deployment in resource-constrained university settings, which our paper aims to address.

6 Conclusion

We present *Marcel*, an open-source RAG-based chatbot designed for handling enrollment-related student questions. The design and architecture targets resource-constrained university settings, taking into account privacy, controllability and observability. We analyze quality-efficiency trade-offs of common RAG-strategies, and provide insights from a real-world deployment at our institution. We plan to improve answer quality through query expansion and to integrate mechanisms for online evaluation of different answering-strategies.

Limitations

We note two limitations of our study. First, we considered only one prompt for the answer generator. It is conceivable that different prompting strategies could be used to improve performance, in particular for deciding when to abstain from answering a query. Second, we have only reported an intrinsic evaluation of the chatbot components. A longer-term evaluation is necessary to see if the system improves information access for students. As mentioned in the introduction, we hope that improved information access leads to a reduction in the number of unsuccessful applications due to formal criteria, and subsequently reduces the workload of academic staff. Understanding whether the chatbot will help to reach those goals requires a long-term evaluation and additional user uptake.

Ethical Considerations

It is important to note that university admissions is a high-risk domain, where wrong chatbot responses may adversarially affect users. At the current stage of the technology, it cannot be guaranteed that the chatbot provides correct answers due to the inherently stochastic nature of LLMs. Therefore, the chatbot only serves as an *additional* information source next to existing student support structures in form of the website, FAQs, email, and phone. To mitigate the risks associated with wrong responses, we (i) insert prominent disclaimers in the user interface, (ii) encourage users to verify important information through the display of relevant sources, and (iii) provide an admin interface to allow periodic review of conversations. A useful extension of this would be automatic flagging of potentially inaccurate responses so that they can be reviewed more efficiently. Lastly, all users in our study gave their explicit consent for the data collection and can retroactively opt-out.

References

- Tom Bocklisch, Joey Faulkner, Nick Pawlowski, and Alan Nichol. 2017. [Rasa: Open source language understanding and dialogue management](#). *Preprint*, arXiv:1712.05181.
- Erin Brereton. 2021. [Higher education’s increasingly nuanced, AI-powered chatbots](#). Accessed: 2025-07-04.
- Sachin Pathiyan Cherumanal, Lin Tian, Futoon M. Abushaqra, Angel Felipe Magnossão de Paula,

- Kaixin Ji, Halil Ali, Danula Hettiachchi, Johanne R. Trippas, Falk Scholer, and Damiano Spina. 2024. [Walert: Putting conversational information seeking knowledge into action by building and evaluating a large language model-powered chatbot](#). In *Proceedings of the 2024 ACM SIGIR Conference on Human Information Interaction and Retrieval, CHIIR 2024*, pages 401–405.
- Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. 2024. [Chatbot arena: An open platform for evaluating LLMs by human preference](#). In *Forty-first International Conference on Machine Learning*.
- Eunsol Choi, Jennimaria Palomaki, Matthew Lamm, Tom Kwiatkowski, Dipanjan Das, and Michael Collins. 2021. [Decontextualization: Making sentences stand-alone](#). *Transactions of the Association for Computational Linguistics*, 9:447–461.
- Gordon V. Cormack, Charles L. A. Clarke, and Stefan Büttcher. 2009. [Reciprocal rank fusion outperforms condorcet and individual rank learning methods](#). In *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 758–759.
- Shahul Es, Jithin James, Luis Espinosa Anke, and Steven Schockaert. 2024. [RAGAs: Automated evaluation of retrieval augmented generation](#). In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 150–158.
- Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. [Precise zero-shot dense retrieval without relevance labels](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1762–1777.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. [The Llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.
- Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, and 196 others. 2025. [Gemma 3 technical report](#). *Preprint*, arXiv:2503.19786.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. [Efficient memory management for large language model serving with PagedAttention](#). In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. [Retrieval-augmented generation for knowledge-intensive NLP tasks](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474.
- Dongfang Li, Xinshuo Hu, Zetian Sun, Baotian Hu, Shaolin Ye, Zifei Shan, Qian Chen, and Min Zhang. 2024. [TruthReader: Towards trustworthy document assistant chatbot with reliable attribution](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 89–100.
- Chin-Yew Lin. 2004. [ROUGE: A package for automatic evaluation of summaries](#). In *Text Summarization Branches Out*, pages 74–81.
- Fengran Mo, Kelong Mao, Ziliang Zhao, Hongjin Qian, Haonan Chen, Yiruo Cheng, Xiaoxi Li, Yutao Zhu, Zhicheng Dou, and Jian-Yun Nie. 2025. [A survey of conversational search](#). *Preprint*, arXiv:2410.15576.
- Long S. T. Nguyen and Tho T. Quan. 2025. [URAG: Implementing a unified hybrid RAG for precise answers in university admission chatbots – a case study at HCMUT](#). In *Information and Communication Technology*, pages 82–93.
- Trung Thanh Nguyen, Anh Duc Le, Ha Thanh Hoang, and Tuan Nguyen. 2021. [NEU-chatbot: Chatbot for admission of national economics university](#). *Computers and Education: Artificial Intelligence*, 2:100036.
- Julius Odede and Ingo Frommholz. 2024. [Jaybot - aiding university students and admission with an LLM-based chatbot](#). In *Proceedings of the 2024 ACM SIGIR Conference on Human Information Interaction and Retrieval, CHIIR 2024*, pages 391–395.
- Onyx. 2025. [Onyx: Open source Gen-AI + enterprise search](#). Accessed: 2025-07-04.
- Cheat Sheet Series Team OWASP. 2025. [Docker security cheat sheet](#). Accessed: 2025-07-04.
- Arjun Panickssery, Samuel R. Bowman, and Shi Feng. 2024. [LLM evaluators recognize and favor their own generations](#). In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
- Xiangyu Peng, Prafulla Kumar Choubey, Caiming Xiong, and Chien-Sheng Wu. 2025. [Unanswerability evaluation for retrieval augmented generation](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8452–8472.

- Nils Reimers and Iryna Gurevych. 2019. [Sentence-BERT: Sentence embeddings using Siamese BERT-networks](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992.
- Stephen Robertson and Hugo Zaragoza. 2009. [The probabilistic relevance framework: BM25 and beyond](#). *Foundations and Trends® in Information Retrieval*, 3(4):333–389.
- Aamir Shakir, Darius Koenig, Julius Lipp, and Sean Lee. 2024. [Boost your search with the crispy mixedbread rerank models](#).
- Weavitate. 2025. [Verba: The golden RAGtriever](#). Accessed: 2025-07-04.
- Hamed Zamani, Johanne R. Trippas, Jeff Dalton, and Filip Radlinski. 2023. [Conversational information seeking](#). *Foundations and Trends® in Information Retrieval*, 17(3–4):244–456.
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. [BERTScore: Evaluating text generation with BERT](#). In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*.

A Benchmarking Rerankers

Retriever	MRR	Δ MRR	R@1	Δ R@1	R@5	Δ R@5	sec/query	Δ sec/query	Params
BM25 + Dense (MARCO)	0.40	—	0.21	—	0.50	—	0.06	—	—
+ Rerank (ms-marco-MiniLM-L12-v2)	0.23	-42%	0.08	-62%	0.35	-30%	0.19	217%	33.4M
+ Rerank (jina-reranker-v1-turbo-en)	0.26	-35%	0.10	-52%	0.40	-20%	0.17	183%	37.8M
+ Rerank (ms-marco-MiniLM-L6-v2)	0.27	-32%	0.11	-48%	0.36	-28%	0.14	133%	22.7M
+ Rerank (jina-reranker-v1-tiny-en)	0.27	-32%	0.12	-43%	0.37	-26%	0.14	133%	33M
+ Rerank (mxbai-rerank-large-v1)	0.37	-8%	0.20	-5%	0.45	-10%	1.59	2550%	435M
+ Rerank (mxbai-rerank-xsmall-v1)	0.42	5%	0.30	43%	0.46	-8%	0.30	400%	70.8M
+ Rerank (mxbai-rerank-base-v1)	0.46	15%	0.33	57%	0.48	-4%	0.59	883%	184M

Table 4: Evaluation of second-stage rerankers. Each metric score is given with its relative delta (Δ) over the baseline. Inference time per query (sec/query) is measured on a GPU.

B LLM Prompts

Listing 1: Prompt for question answering based on retrieved documents.

```
// System prompt
You are a helpful and engaging chatbot called Marcel. If someone asks you, your name is Marcel
and you are employed at the Marburg University. You answer questions of students around their
studies. Please answer the questions based on the provided documents only. Ignore your own
knowledge. Don't say that you are looking at a set of documents. If you cannot find the answer
to a given question in the documents you must apologize and say that you don't have any
information about the topic (e.g., "Unfortunately, I do not have any knowledge about <rephrase
the question>").

// (Optional) Conversation history (turns of user/assistant messages)
...

// User prompt for question answering
Given these documents, answer the question.

## Documents
{% for doc in documents %}
### {{ doc.title }}
{{ doc.content | replace("\n", "\\n") }}

{% endfor %}

## Question
{{ question }}
```

Listing 2: Prompt for question answering without retrieval.

You are a helpful and engaging chatbot called Marcel. Your name is Marcel and you are employed at the Marburg University. You help students with questions regarding admission and enrolment to the MSc Data Science Program. Only engage in topics related to admission and enrolment. Do not engage in other discussions.

Listing 3: Prompt to classify a query into retrieval or no-retrieval.

Please determine if the following user utterance is a question. Respond with 'YES' if it is a genuine question. Respond with 'NO' if it is chit-chat or if the user asks the chatbot what kind of information it could provide, unrelated to earlier conversation. Only respond with the label YES or NO.

Message: {message}

Listing 4: Prompt to detect non-answers for evaluation of selectivity.

```
## Instruction
Classify the given response as either an answer or a non-answer.

Non-answers are responses that indicate a lack of information or inability to provide an
answer. They typically include phrases such as:
- "I don't know..."
- "I can't provide information about..."
- "It is not stated..."
- "There is no answer"
- "There is no information..."
- Or other synonymous expressions indicating uncertainty or lack of information

Any response that provides concrete information or a direct answer, even if brief, should be
classified as an answer. For each response, provide a classification in the following format:
Classification: {'rationale': 'Brief explanation of why it's an answer or non-answer',
'non_answer': 0 or 1}

Use 0 for answers and 1 for non-answers in the 'non_answer' field.

## Examples
Response: The information provided does not mention anything about compulsory elective modules
or the number of free modules a student can choose from in the M.Sc. Data Science program at
Marburg University.
Classification: {'rationale': 'The response indicates a lack of information, which is
characteristic of a non-answer.', 'non_answer': 1}

Response: The question is not clear because there is no specific module mentioned. The text
only talks about enrollment as a doctoral student and provides information about the required
documents and the enrollment process. It does not mention a specific module.
Classification: {'rationale': 'The response states that specific information is not provided,
making it a non-answer.', 'non_answer': 1}

Response: Master of Science (M.Sc.)
Classification: {'rationale': 'The response provides a specific piece of information, making
it a valid answer.', 'non_answer': 0}

Now, classify the following response:
Response: {{ response }}
Classification:
```

C Example FAQ Data

#	FAQ	Source
1	Which Bachelor degree is needed to be admitted to the Masters of Data Science program?	(D ₁) Master Data Science Admission Requirements; (D ₄) Master Data Science Eligibility Assessment Process
2	Which language certificates are accepted?	(D ₁) Master Data Science Admission Requirements; (D ₅) Demonstrating English proficiency (C1); (D ₆) Demonstrating English proficiency (B2); (D ₇) Demonstrating German proficiency
3	What are the application deadlines?	(D ₈) Application Deadlines for Master Programs

Table 5: Example FAQ data. Each question is linked to answer-relevant documents in the knowledge base.

D Example Evaluation Data

#	Question	Reference Answer	Source
1	I am writing to ask whether me a Aerospace Engineering graduate can apply for the Master in Data Science? If yes will there be additional requirements?	Your bachelor degree needs to be in Data Science, Mathematics or Computer Science or comparable. Overall, at least 90 credit points (CP) must have been completed in academic courses on Mathematics and Computer Science. Note that there is a list of specific courses that are required, and you need to cross-check whether you fulfil the requirements on those, too. Industry training or professional experience does not count as academic education.	(D ₁) Master Data Science Admission Requirements
2	I have 2 questions: Is German language A2 required during online application? If I don't have proof of German language A2 level, Is it possible to be accepted?	You need to prove either English level C1, or German DSH-2 and English B1. Proof of language proficiency needs to be submitted with the application.	(D ₁) Master Data Science Admission Requirements; (D ₂) Application with a degree from abroad (first semester)
3	I aspire to take admission at your esteemed university for masters in data science for second semester in summer as I am pursuing masters in data science for first semester at a university in Germany currently for the winter semester. I wanted to know how to apply for second semester and the deadline for it.	In short, you need to obtain an advanced placement certificate from the departmental advisor of the Masters Data Science programme. You then and apply via uni-assist, indicating an application for the second semester. The process is explained in detail on the university website.	(D ₃) Application with a degree from abroad (higher semester)
4	Can my application be checked before upload to see whether I am eligible. I do not want to pay the application fee if I would not be admitted to the program.	The admission committee cannot do any pre-application checks. You can check your transcript of records yourself and see whether it matches the admission criteria.	n/a
5	I am now in the process of preparing my papers for submission, but due to some circumstances, it may be a little delayed. In the end, everything will be ready before Deadline on February 23. My question is: Does delaying the application affect the chance of acceptance? Does the acceptance stage take place during the application period or does it not take place until after the deadline has passed for all applicants?	Applications are checked independent of each other, so your chance of admission does not depend on the time you submitted the documents. It is only checked whether the admission requirements are fulfilled. Offer letters will be sent as soon as the application is checked.	n/a
6	If I am unable to obtain my visa before the semester begins, would it be possible to defer my admission to the next semester master data science?	You can transfer your admission to the next semester one time if the following conditions are met: (1.) You are not yet enrolled. (2) The study program is offered in the following semester (which will be for Master Data Science). (3) Admission Conditions in the Study Regulations have not changed (not planned for Master Data Science). You should then write an email to masters@uni-marburg.de in December / July and request the transfer. Then you will get a new letter of admission (after the check whether the above conditions are met).	n/a

Table 6: Sample questions from evaluation data. Each question has a manually written reference answer. Questions that are answerable with the knowledge base (#1–3) also have ground-truth sources to evaluate retrieval. Unanswerable questions (#4–6) are used to evaluate generator’s ability to abstain.

E Example Generations Reflecting Current Limitations

User: I have an industrial engineering bachelor degree from iran but my requirement courses are 50% related to requirement of data science program can I be eligible?

Marcel: According to the Data Science (Master of Science) document, it is stated that "We require proof that you have completed a relevant Bachelor's degree in the fields of Data Science, Mathematics or Informatics or proof of a comparable domestic or foreign university degree qualifying you for a profession."

Additionally, it is required that "at least 72 credit points (CP) must have been earned in modules from the scientific discipline of Computer Science and at least 18 CP in modules from the scientific discipline of Mathematics."

However, it is not explicitly stated that the entire undergraduate degree program must be directly related to Data Science. Nevertheless, it is stated that "It is strongly recommended that students take responsibility for acquiring the skills from the above modules that they have not already acquired before commencing their studies."

Since you have mentioned that only 50% of your requirement courses are related to the requirements of the Data Science program, it might be possible that you are eligible, but you would likely need to demonstrate that you have acquired the necessary skills in the areas where your undergraduate degree program was lacking.

You can contact Prof. Dr. Christin Seifert for content-related questions about the program, and Master's Coordination for formal questions about your application. Email: christin.seifert@uni-marburg.de Email: master@uni-marburg.de

User: about eligibility assessment does operations research is related to 'Basic Real Analysis' or 'Analysis I'?

Marcel: Unfortunately, I do not have any information about Operations Research being directly related to 'Basic Real Analysis' or 'Analysis I' in the context of the Data Science program at Marburg University in the provided documents.

User: so what are fundamentals of basic real analysis

Marcel: Based on the provided documents, the content of "Basic Real Analysis" is not explicitly stated. However, "Analysis I" and "Analysis II" are mentioned as alternatives to "Basic Real Analysis" in the prerequisites for some modules.

In the module "Stochastic Analysis", it is mentioned that either "Analysis I and Analysis II" or "Basic Real Analysis" is recommended. This suggests that "Basic Real Analysis" may cover similar topics as "Analysis I and Analysis II", but with a focus on real analysis.

In general, a course on real analysis typically covers topics such as:

- Real numbers and their properties
- Sequences and series of real numbers
- Continuity and differentiability of real-valued functions
- Integration of real-valued functions
- Metric spaces and topology

However, without more specific information, it is difficult to say exactly what topics are covered in the module "Basic Real Analysis" at Marburg University.

User: what about Efficient Algorithms

Marcel: Efficient Algorithms is a module that is part of the Data Science program at Marburg University. It is mentioned as one of the recommended prerequisites for the Parameterized Algorithms module, and it is also mentioned as an example of a module that students may take as part of their studies in the Perspectives document.

According to the Online-Modulhandbuch, the "Efficient Algorithms" module (CS 566) is an advanced module that covers topics such as algorithmic techniques, greedy algorithms, dynamic programming, and space and time complexity. The module has a workload of 270 hours and is worth 9 credit points.

The qualification goals of the "Efficient Algorithms" module include:

- Applying skills in designing algorithms and knowledge of key design and analysis paradigms
- Using efficient data structures in algorithm design
- Analyzing algorithms with respect to correctness and effort
- Practicing scientific working methods (recognizing, formulating, solving problems, abstraction)
- Training to speak freely about scientific content

There are no explicit prerequisites for the "Efficient Algorithms" module, but the following modules are recommended:

- Algorithms and Data Structures
- Practical Informatics II: Data Structures and Algorithms for Pre-Service-Teachers

The module is assigned to the Computer Science department and can be attended by students in various study programs, including B.Sc. Data Science, B.Sc. Computer Science, M.Sc. Data Science, M.Sc. Computer Science, and others.

Figure 2: Example where a user wants to determine if their academic background meets admission requirements.

User: For Data Science can I submit my IELTS after deadline?

Marcel: Unfortunately, I do not have any information about the possibility to submit IELTS scores after the deadline for the Data Science Master's program.

User: Can I submit my APS after deadline?

Marcel: Unfortunately, I do not have any information about the possibility to submit APS scores after the deadline for the Data Science Master's program

Figure 3: Examples where the chatbot abstains from answering due to knowledge gaps.

User: Is a Thesis required for admission to msc data science?

Marcel: No, a thesis is not a requirement for admission to the Data Science Master's program. However, if a final thesis worth at least 10 credit points has already been completed in the fundamental university degree program, proof of this final thesis and its scope must be submitted as part of the eligibility assessment process.

User: Is english required

Marcel: Yes, English proficiency of at least level B1 of the "Common European Framework of Reference for Languages" or English proficiency of at least level C1 is required, depending on the specific circumstances.

User: What are the circumstances

Marcel: It appears that the circumstances refer to the situation in which initial applicants can move into an apartment straight away. According to the text, only under individual circumstances can initial applicants move into an apartment straight away, and in such cases, an early application is strongly advised.

Figure 4: Example of a contextualization issue. The third query is a follow-up on the second query. Omission of the subject ("circumstances when B1 or C1 is required") lead to retrieval/generation of unrelated information.

User: Hi Marcel, what to study at uni marburg?

Marcel: You can find explanations of our degrees in the "Studying in Marburg" section, which may help you get a first impression of life in Marburg and the variety of subjects we offer.

User: Can you show me a link to that?

Marcel: As a chatbot, I don't have the capability to share links. However, you can visit the University of Marburg's official website, and from there, you can navigate to the "Studying in Marburg" or the "Degree programmes" section. This will give you more information about the various courses they offer, including the MSc Data Science program that I can help you with.

Would you like to know more about the MSc Data Science program specifically?

User: Why should I study Msc. DataScience?

Marcel: Unfortunately, I do not have any information about why you should study M.Sc. Data Science specifically, but you can find an overview of the program on the [M.Sc. Data Science web page](#), which includes the required study skills and interests, as well as the content and structure of the program.

Figure 5: Example of a link-seeking question. The generator cannot embed links to sources in the main text (second answer), unless some retrieved context happens to mention the link (third answer).

