

Be safe, don't hallucinate! Developing an AI agent for MR safety requests.

Oscar Lally, Marc Miquel

MRI Physics, Medical Physics and Clinical Engineering, Guy's and St Thomas' NHS Foundation Trust, London, United Kingdom



Background and Aim:

Radiology departments are under more strain and ever, and one of the root causes of this is the necessity to scan an ever-increasing number of implants. Ascertaining MRI conditions to scan implants often requires time-consuming searches through manuals, research papers and websites. This additional workload and operational stress placed on radiology departments is often considered an inevitability, but using machine-learning techniques can reduce the workload to find MRI conditions drastically. **We propose a method to streamline this process by automatic retrieval of information, and generation of preliminary advice.**

Framework:

Most MRI conditional implants have manuals that specify the scanning conditions. However, these documents tend to be very long (up to 100 pages), they tend to lack structural clarity and can they sometimes be intentionally difficult to interpret. We propose a simple user-interface that makes use of a **Retrieval-Augmented Generation (RAG)** model to generate an MR safety report. A RAG is an AI framework that combines the benefits of a generic Large Language Model (LLM) output with highly specific, selected data such as MRI manuals. The result is an application that incorporates the flexibility and familiar interface of an LLM combined with a highly restricted output that is forced to only give answers based on the data fed to it.

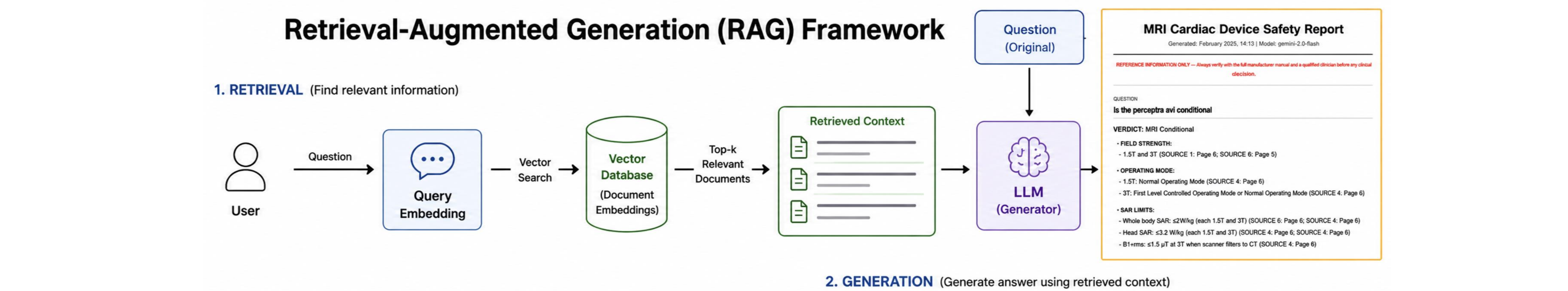


Figure 1. Diagram explaining the two step process of a RAG pipeline. Retrieval involves embedding the MRI manuals into a vector database, and generation uses a pretrained LLM combined with the embedded vector database to give replies strictly limited to MRI Safety.

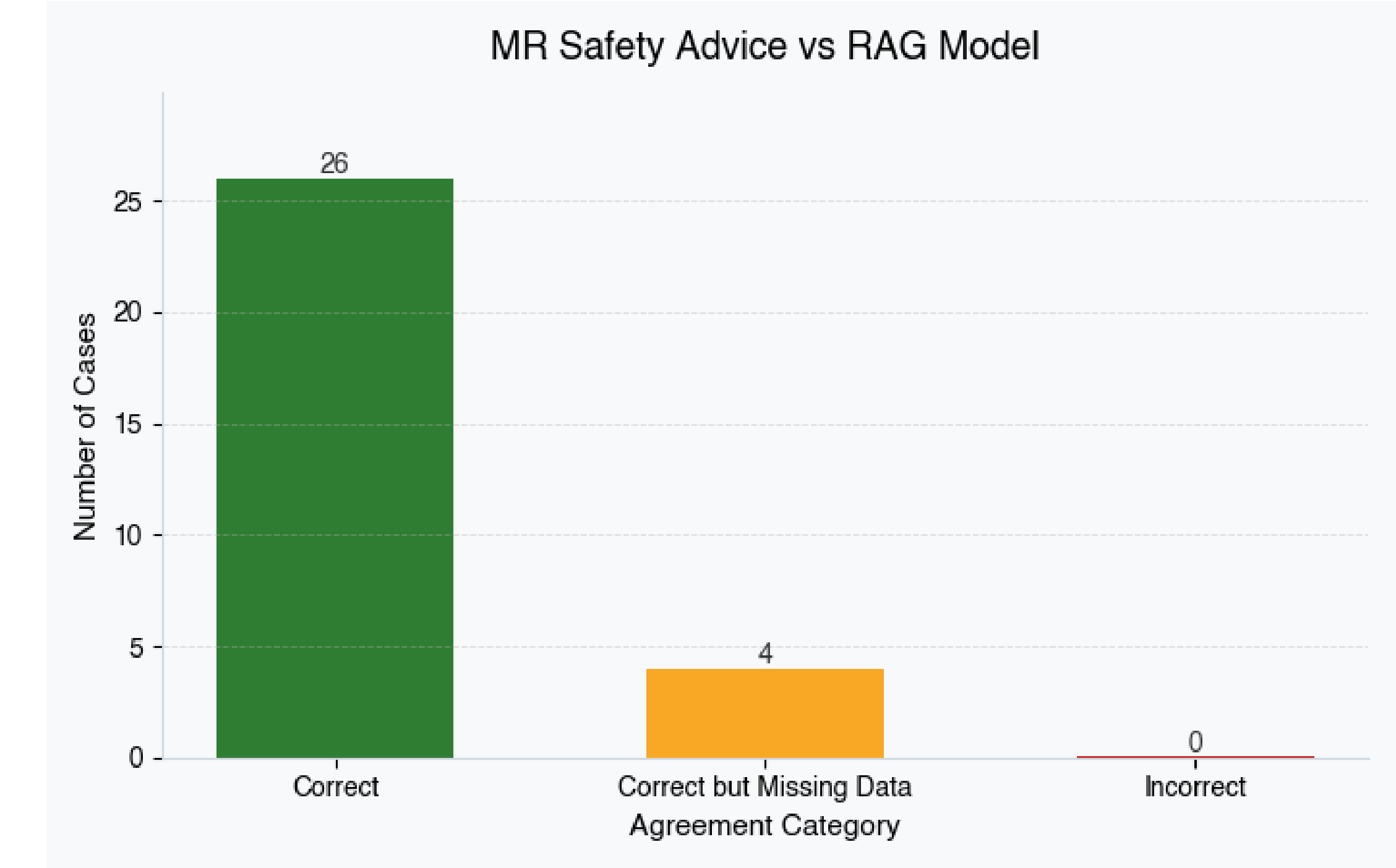


Figure 2. Bar chart demonstrating comparisons between the responses between MRSEs and the RAG model. The model produced 0 incorrect answers and 26 direct matches with the advice given by the MRSEs. 4 responses were correct in that the implants were conditional, but they missed one or more specific conditions.

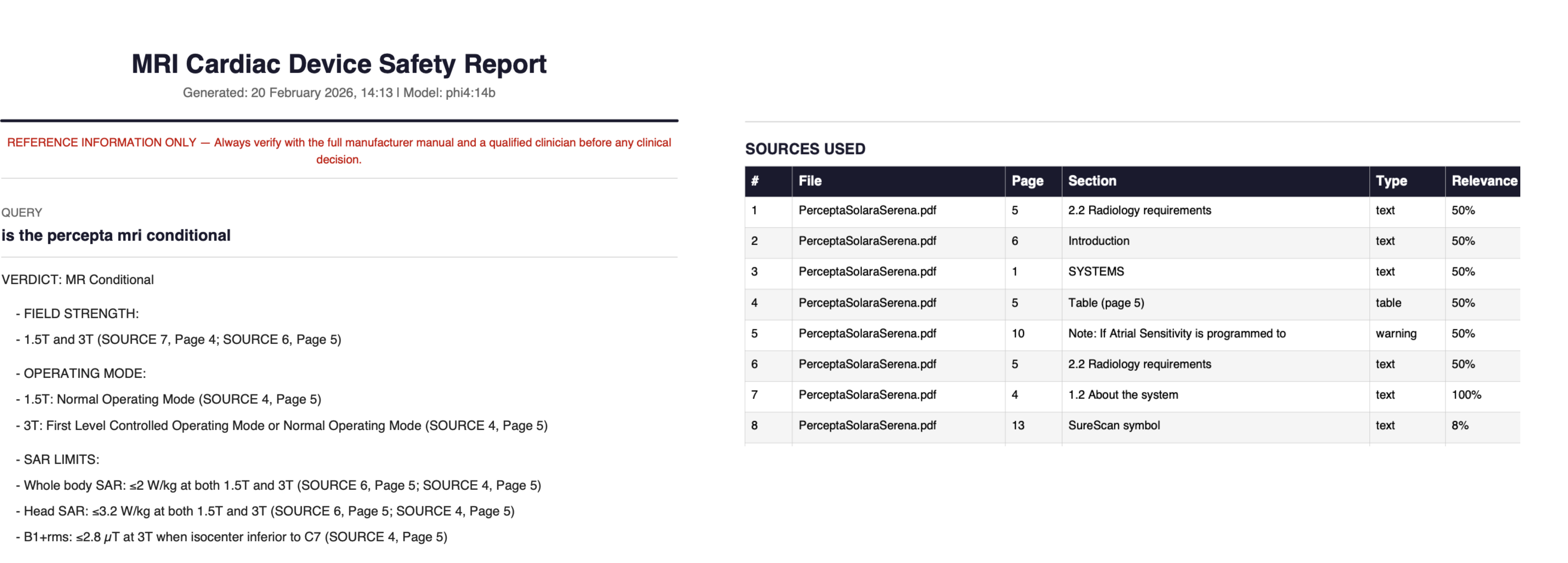


Figure 3. Snippets from a generated report to be given to a radiographer/radiologist. The report contains the important information regarding the MRI conditions of the implant. It also shows exactly what document, page, section and type of information that the presented conditions were drawn from.

Technical Details:

Manuals from over 100 implants were downloaded and organised by manufacturer and then implant type. All software was created using Python. The pdfs were parsed using the multi-faceted pdfplumber library, allowing for text, tables other more complicated structures within the pdf to be parsed cleanly and converted to text. The query embedding was done via the mxbai-embed-large embedding model. This model is a high-performance, open-source embedding model developed by mixedbread.ai. The model converted the text extracted via pdfplumber into 1024-dimensional vectors that represent the text's semantic meaning.

The generator used was the llama 3.1 model. Once running, llama 3.1 hosts a local server allowing other apps (such as the aforementioned Python embedding script) to communicate with it. The generator was forced to use a reply template that limited the range of outputs it could produce. Despite being less 'human', this restriction forced the model to return the implant's MRI conditions every time any MRI related question was asked, regardless of how the question was phrased. In order to optimise between accuracy and time, the top 20 relevant document chunks were identified using a more general but very quick filter, and then a far more specific filter was used to re-rank the original 20 document chunks to find the best 10 chunks that would then be used to inform the RAG's response. The level of allowed hallucination was set to 0, ensuring that the model could only use document chunks to inform its answer, enhancing the safety and reliability of the application. Reports based on the responses are created using the reportlab and platypus libraries on Python. They include both scanning conditions and which part of the documents were used to generate the report

Tests and Results:

30 safety requests about implant conditions, including 15 Implantable Cardioverter Defibrillators, 10 Spinal Cord Stimulators and 5 Cochlear Implants were used to assess the reliability of the outputs of the model. Responses from a Magnetic Resonance Safety Expert (MRSE) were directly compared to the model's output (Fig. 2).

Conclusions:

Overall, the RAG model developed has potential to almost completely automate MRI safety queries. The resulting time saved can bolster the ability for physicists to do research, radiographers to scan more patients, and radiologists to work on other cases, enabling more patients to be treated effectively.