POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Report: developing a conversational agent to explain CNN-based pneumonia detection to medical sudents

Project for the course on Human-Computer Interaction for Artificial Intelligence

Authors: **Marco Gianvecchio**
Giacomo Da Re
Lorenzo Manoni

Academic Year: 2022-2023

Contents

1 | Introduction

The field of human-computer interaction (HCI) and its applications are exponentially increasing in importance. In parallel with the massive use of artificial intelligence (AI) systems in everyday life, the ways and quality in which a user can interface with these systems should improve at the same pace. HCI and AI have often struggled to dialogue, despite having common interests based on the connection between machines and intelligent human agents [1]. The need for user-facing AI systems that are reliable, safe, and trustworthy has pushed engineers and designers towards giving control back to the users, instead of presenting them with "black boxes" that are difficult to understand.

In this project we explore, through the development of a simple chatbot, how a continuous interaction between machine and user can actually help the latter to complete their task intuitively. We start with a convolutional neural network (CNN) that can assist the diagnosis of pediatric pneumonia from X-Ray images and design around it a conversation that could help medical students improve their skills. Like an experienced mentor, the chatbot guides the student toward a better understanding of each case. Thanks to this interaction paradigm, the user is actively confronted with the model, learning about it and how to use it in the way that is most useful for them.

2 | Materials and methods

2.1. Data preparation

The images and data we show to the medical student during the interaction with the chatbot are all based on a trained artificial neural network.

2.1.1. Model selection

The project guidelines required us to start from an existing CNN architecture available as a code example for the Keras library [2]. This allowed us to focus on the transparency, interpretability, and explainability of the model, instead of aiming to improve its accuracy. The network's purpose is to diagnose pediatric pneumonia using chest X-ray images. The training images from Cell [3] are already classified and freely available for download, they are also split between training data and test data. Once trained the CNN can take as input a new X-Ray and output a numeric value between 0 and 1, if the value is higher than 0.5 then the prediction is that the patient has pneumonia.

Since our conversational agent is aimed at helping medical students, we thought it would be appropriate to show them only X-ray scans of which we know the true label. This way we can use the model's prediction to guide the student toward a better understanding of the diagnosis, without the risk of showing them a false positive or false negative. Therefore when showing a heatmap or similar images during the conversation we made sure that the source X-Ray both belonged to the training set and was correctly classified by the model. In other words, we never present the student with an image on which the model has made a wrong prediction.

2.1.2. Training and testing

The training set consists of 3883 chest X-Rays with presence of pneumonia and 1349 without. Class weighting was used before training to correct this imbalance in the data.

We fixed the random seeds for NumPy and TensorFlow to make our results more reproducible. The model was then trained using an NVIDIA T4 GPU included with Google Colab's free plan, as the TPU was unavailable. Training took less than 7 minutes, stopping at epoch 26. We then evaluated the model on the test set, obtaining 73% binary accuracy, 70% precision, and 99% recall.

2.1.3. Heatmap generation

One way the student could be assisted during the learning process is if instead of just being given a label (this patient does or does not have pneumonia) they could see where in the image they should focus on to make the correct diagnosis.

This can be achieved using a heatmap that shows what area of the images the trained neural network considers more important to make its prediction.

Several methods were considered for generating the heatmap but in the end Grad-CAM was chosen as it is well known and has been successfully applied in cases analogous to ours [4][5].

We sourced a Grad-CAM implementation from Keras [6] and applied it to the X-Ray scans. The only parameter we had to set was the name of the last convolutional layer, which the heatmaps are based on.

Additionally, given the textual nature of the conversational agent, we discussed adding a written explanation of the heatmap. This was achieved using OpenCV, a computer vision library. Our code simply identifies the red points in the heatmap (red means very important according to the CNN) and then averages them. The average point is then converted to a textual description of the area of interest for the student like "top left".

2.1.4. Image embedding for similarity

If the student has a hard time understanding why a certain X-ray shows presence or absence of pneumonia, they might benefit from seeing similar scans with the same diagnosis.

We discussed various ways to relate images to one another. Applying a traditional method like mean-square error directly on the original X-Rays would discard all the precious information our model has learned. Instead, we thought of simply using the output of one of the last dense layers of the CNN as an embedded representation of each image. A vector of size 512 is therefore saved to disk for every X-Ray scan, and when we need to find N images related to a given one we simply apply SciPy's cosine similarity and find the N closest vectors.

We tested this method and empirically verified that it seems to select images showing the same area of interest, highlighting for example that the model has focused on the bottom left area of N+1 X-Rays to classify them as normal.

2.2. Conversational agent

Conversational agents are a particular application of Artificial Intelligence for chatbots. This kind of technology implements Natural Language Processing techniques in order to understand and respond to human-spoken queries. Recently this technology has become really popular as a more user-friendly alternative to traditional User Interfaces.

Chatbots are really simple to use even for people that are not into technology, that can find UIs too complex and difficult to navigate. Chatbots provide an intuitive solution to users that can type or speak their intentions and the chatbot will provide a real-time response reducing time and effort for task completion. This makes the process of using the application more engaging and satisfying.

Nowadays chatbots are really easy and cheap to develop, and this reason and those above led to an exponential use of chatbots in different kinds of applications, such as healthcare, e-commerce, and customer service, and are likely to become even more popular in the future.

Early on during development we came up with the diagram shown in Figure 2.1. This was very useful while building and later testing the chatbot, as it shows all possible ways the conversation could go.

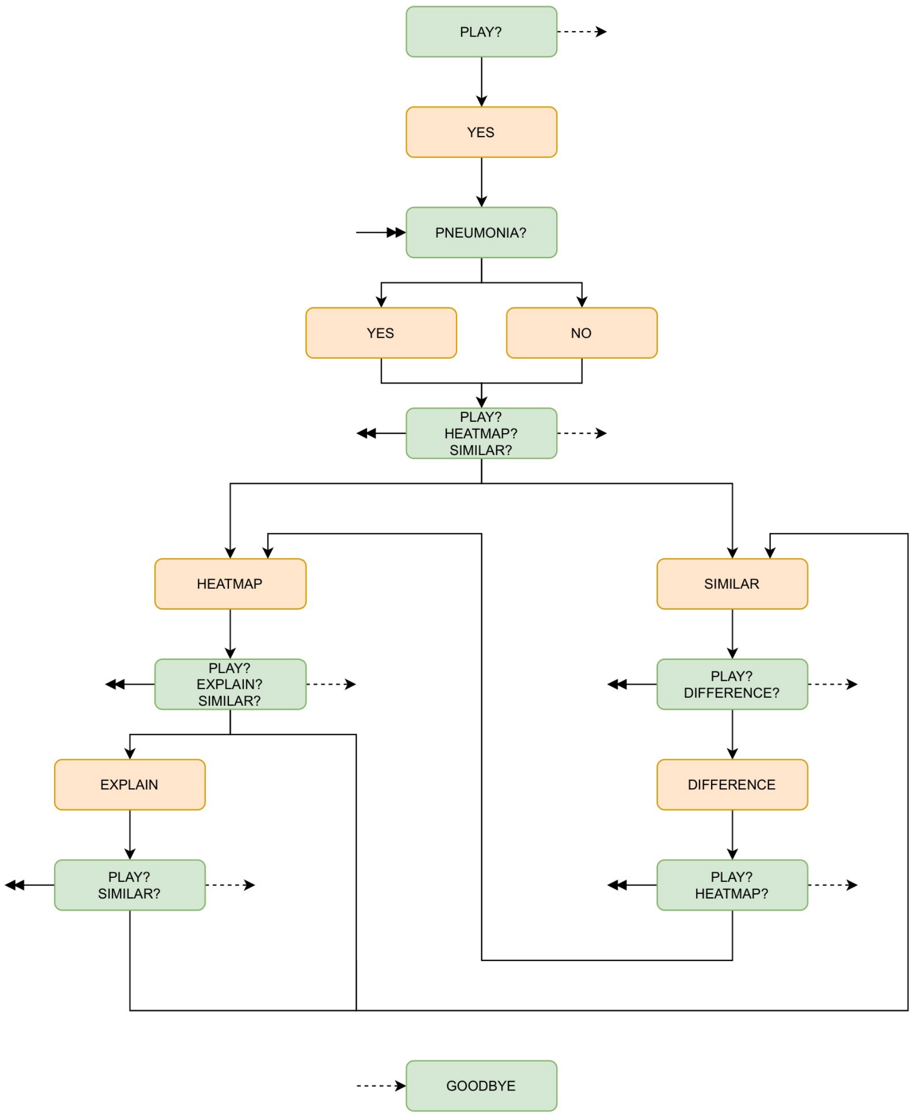


Figure 2.1: All possible interactions between the student and the chatbot. Each rectangle summarizes words spoken by the agent (green) or the user (orange). Dashed arrows represent the end of the conversation (e.g. the user says "I want to stop" and the chatbot responds "Goodbye").

2.2.1. Dialogflow CX

We developed our conversational agent with Dialogflow CX in order to simulate a professor/medic that gives indications to the med student in a more discursive way than traditional UI. Dialogflow is a powerful tool for building conversation chatbots, we used CX instead of ES because it offers a wider range of capabilities, and we just use a small portion of its features. A particularly helpful tool is the "minimap", reported in Figure 2.1. It shows at a glance the flow of the conversation.

We were been able to train the chatbot to understand the intentions of the user and respond to a wide range of inputs, in order to guide the user into all the functionalities of the application. This allowed us to create a chatbot that could provide accurate and relevant information to users, while also being easy and intuitive to use.

We then integrated the chatbot into our application through its APIs and gave the user the appropriate output for each intent of the pneumonia chatbot. Dialogflow CX integrates robust analytics and reporting tools, which allowed us to debug the performance of our chatbot and make improvements over time, ensuring that our users always receive the best possible experience.

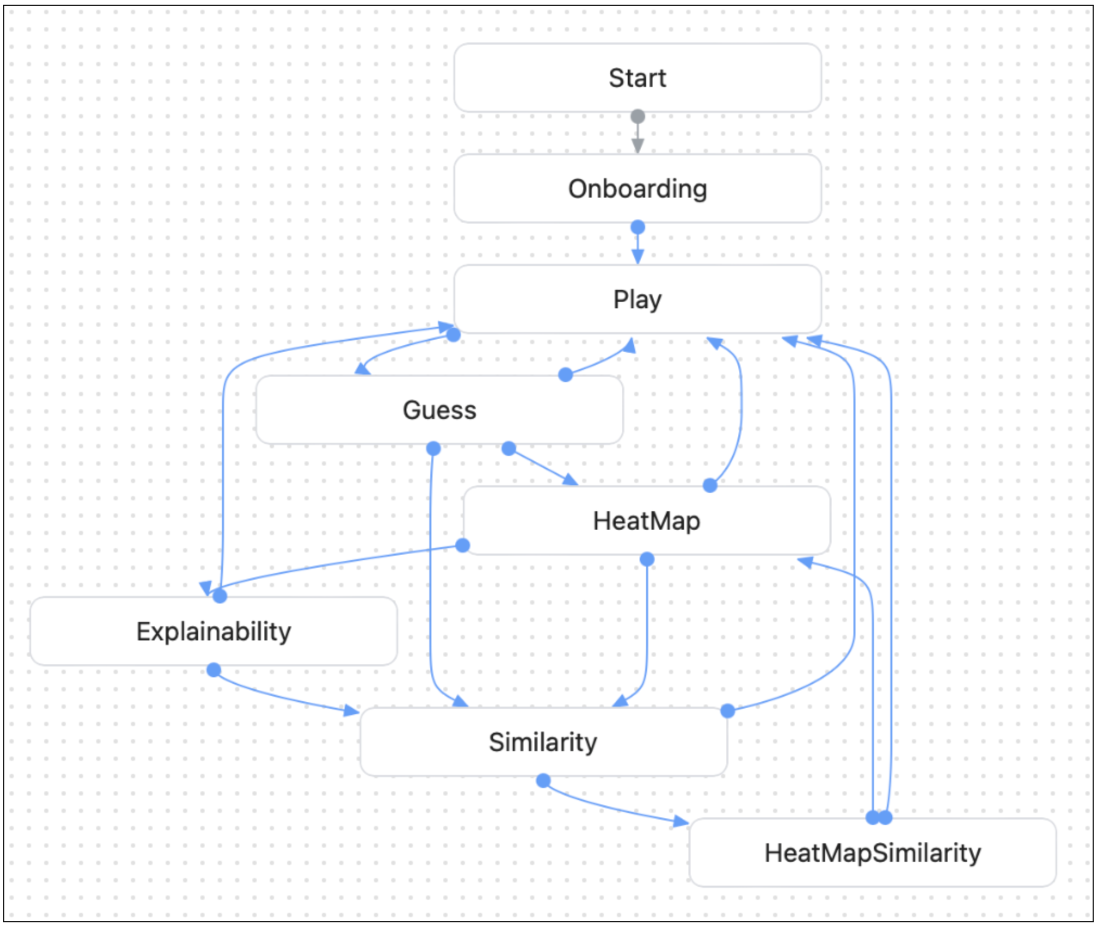


Figure 2.2: Map of the Pages as shown by Dialogflow.

2.2.2. Improving user interactions

Starting from the code provided during the lectures we gradually applied modifications to tailor the interaction to the needs of our specific application. On the user interface side two changes were made: adding a button to mute the chatbot (as it's possible the student could prefer just typing) and implementing a way to display images in the chat without interrupting the conversation.

We also carefully added elements to the conversation we thought would add value and improve the interaction. Several choices were made along the way, these are the most significant:

- The user is greeted by a welcome message, explaining the purpose and method for the interaction.
- When shown similar images the student is also told how similar they are. Medical students also study statistics and might benefit from a precise numerical approach. This value is directly derived from the cosine similarity function.
- The chatbot is very loquacious, actively presenting to the user possible ways to interact, so the conversation never hangs.
- After asking for similar images, the user can also ask for their heatmaps, to better understand why they were identified as comparable by the model.
- If the student has difficulty understanding the heatmap they can ask for a textual explanation, which will tell them how to read it and where to look.
- To make the conversation more dynamic and less repetitive the chatbot randomly alternates between different ways of saying the same thing (e.g. it will sometimes propose to "display a heatmap" and sometimes to "show the area of interest" which produces the same result).

3 Results

3.1. Example conversations

Figure 3.1 and Figure 3.2 (spanning across 2 pages) show some of the capabilities of the conversational agent. The words typed by the chatbot, shown as grey bubbles on the left, are also spoken out loud using text-to-speech synthesis. The two buttons on the bottom right allow the student to mute the chatbot and to speak their request instead of typing it down.

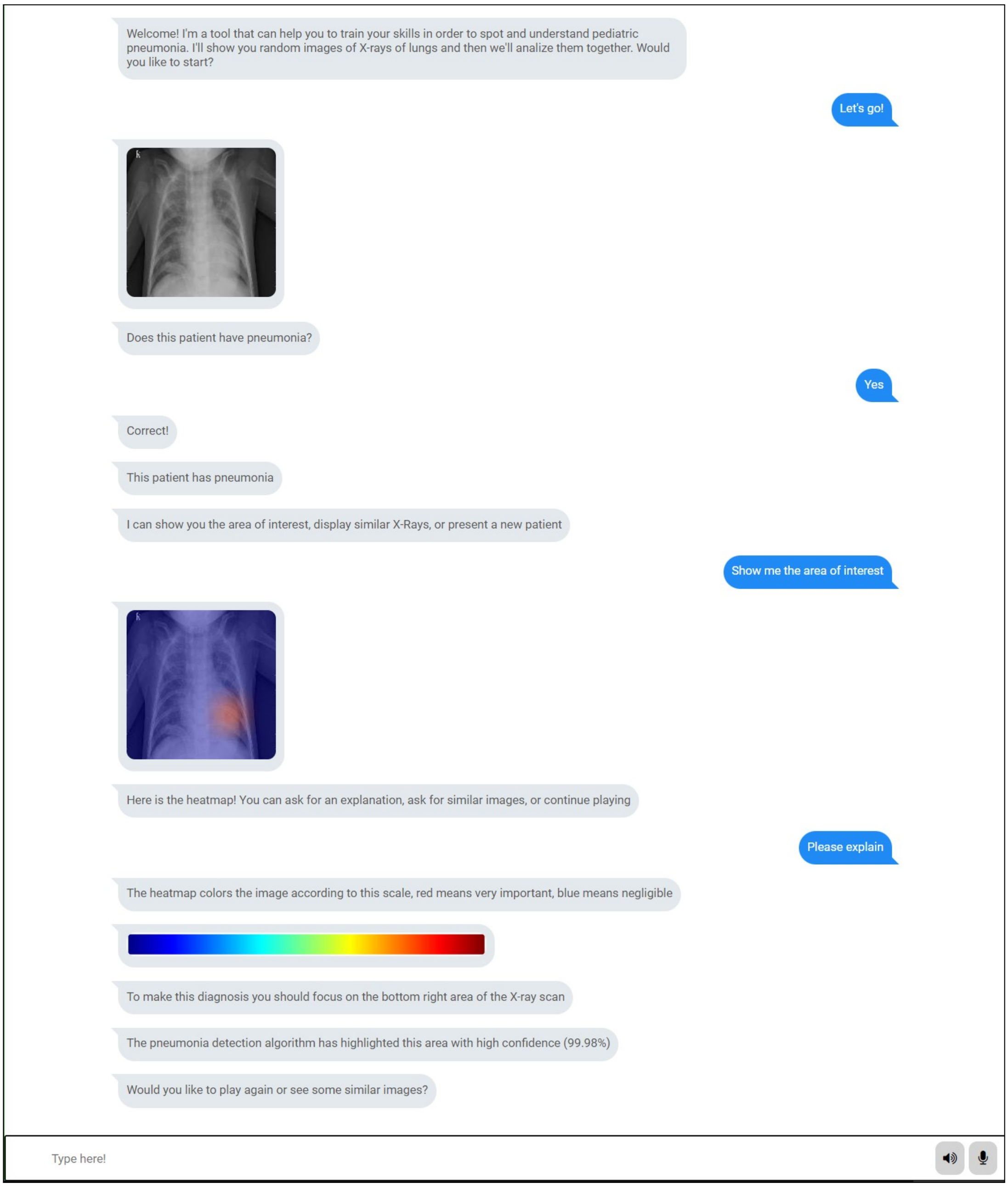
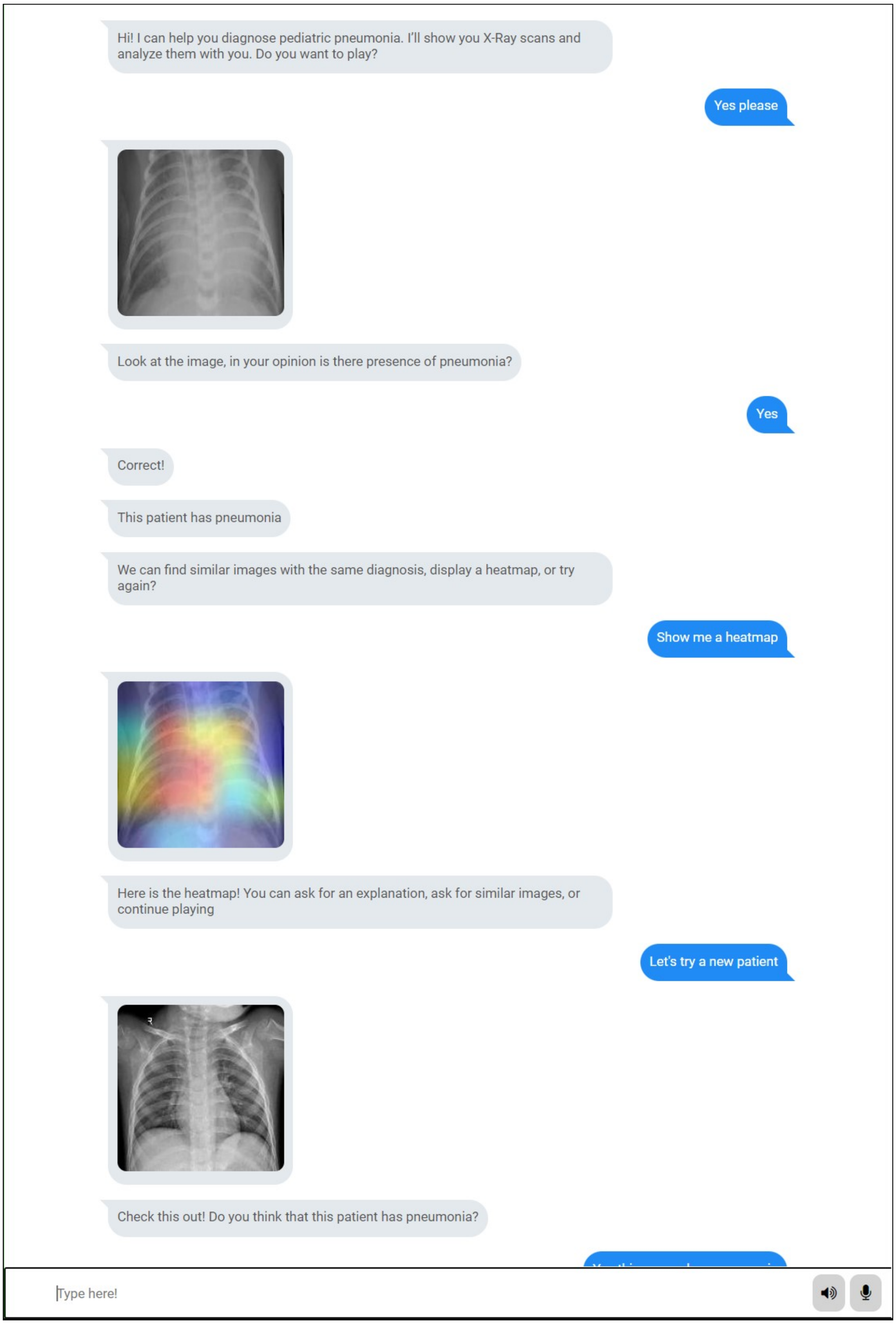


Figure 3.1: A conversation with the chatbot.


Hi! I can help you diagnose pediatric pneumonia. I'll show you X-Ray scans and analyze them with you. Do you want to play?
Yes please
Look at the image, in your opinion is there presence of pneumonia?
Yes
Correct!
This patient has pneumonia
We can find similar images with the same diagnosis, display a heatmap, or try again?
Show me a heatmap
Here is the heatmap! You can ask for an explanation, ask for similar images, or continue playing
Let's try a new patient
Check this out! Do you think that this patient has pneumonia?
Type here!

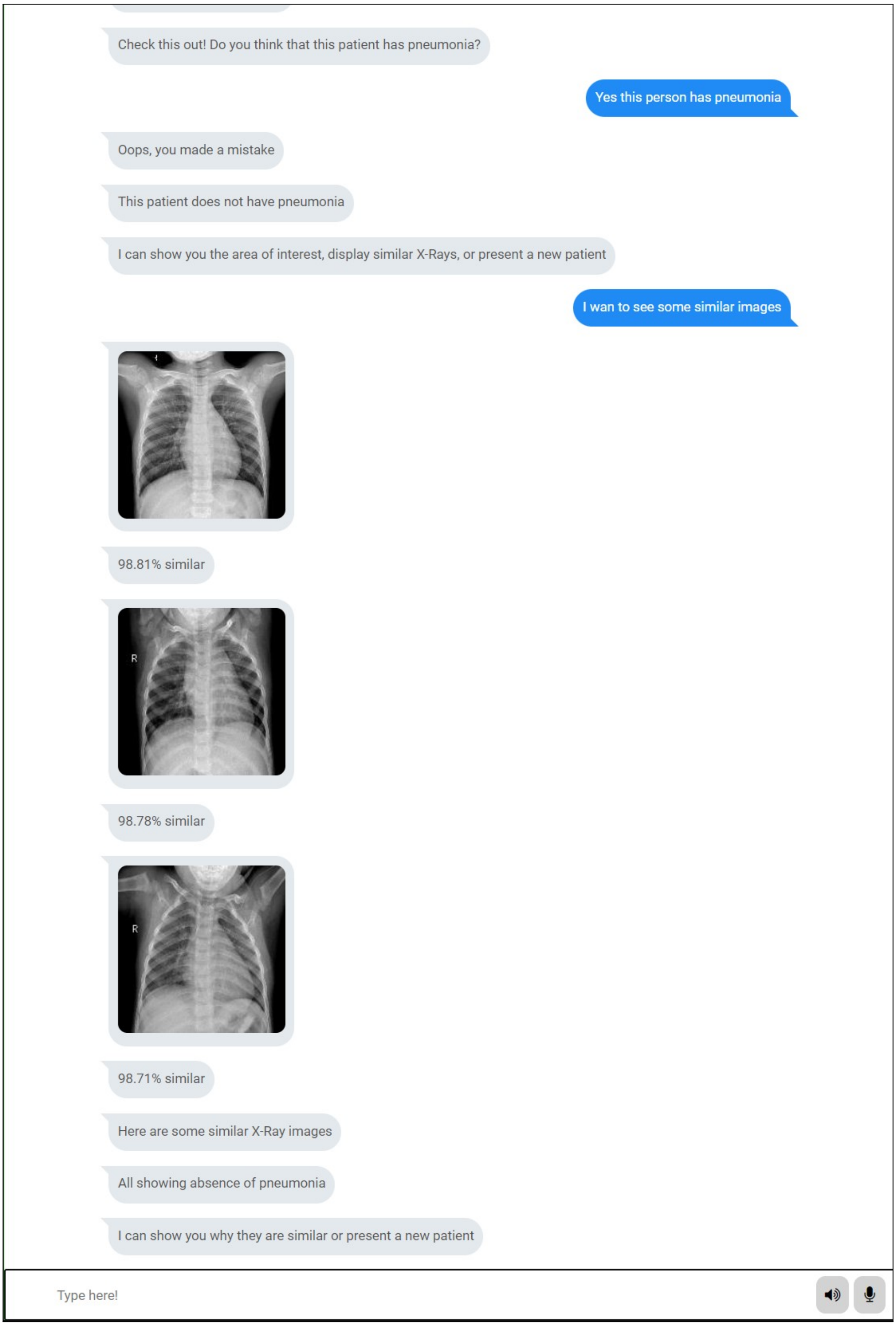


Figure 3.2: Another example conversation.

4 | Discussion

Although our chatbot works as intended it was developed as an experiment to explore the transparency, interpretability, and explainability of AI models, and is therefore not intended as a real product. We discuss some of the limitations of this project as well as improvements that we would have liked to work on with more time and resources.

The design and development were carried out without the supervision or guidance of any medical professional. Consulting experts on the matter would have allowed us to make sure that the data points the chatbot focuses on and the information it provides are accurate and useful. Furthermore having a test group composed of medical students would have provided helpful real-world feedback, instead we had to rely on the opinion of a few friends who study medicine.

The CNN we used as a basis for information extraction is 73% accurate, we noticed this is lower than the state of the art [5] and therefore the amount and quality of data we derive from it are improvable. The heatmaps, which are derived from the activations of the neurons in the network, in some cases highlight irrelevant areas (e.g. outside of the chest of the patient). Furthermore, the prediction confidence is almost always high (rarely dropping below 99%).

The focus of the project was on human-computer interaction and explainability, however it's possible that the learning process for the student would have benefited from a more accurate model. Several improvement paths could be tried, from fine-tuning the CNN to selecting a completely different model that provides richer information than just binary classification.

5 | Conclusion

The chatbot we have proposed and developed is not only the result of the last month of work but is the synthesis of the entire journey that began in February with the panels and lectures proposed by the professors. Without them and all the theoretical background provided to us through slides, code, and in-person meetings the development of the conversational agent we have described would not have been possible.

This was not a mere technical exercise for us, but the synthesis of all the theoretical knowledge gained during the course that resulted in the realization of our work. As master's students, the problem of explainability was never placed in the spotlight during the course of study, but the basics we acquired on Human-Computer Interaction for Artificial Intelligence will surely serve as a basis for further study.

6 | Availability of data and materials

All the source code and results from the data preparation phase, as well as the actual application, are made available on GitHub [7].

7 References

1. Green S, Heer J, Manning CD. Natural Language Translation at the Intersection of AI and HCI: Old questions being answered with both AI and HCI.
2. `https://keras.io/examples/vision/xray_classification_with_tpus/`
3. Kermany DS, Goldbaum M, Cai W et al. Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning.
4. Panwar H, Gupta PK, Siddiqui MK et al. A deep learning and grad-CAM based color visualization approach for fast detection of COVID-19 cases using chest X-ray and CT-Scan images.
5. Kundu R, Das R, Geem ZW et al. Pneumonia detection in chest X-ray images using an ensemble of deep learning models.
6. `https://keras.io/examples/vision/grad_cam/`
7. `https://github.com/rtjk/hci4ai-project`