

Prototype Documentation

Why rule-based?

These systems are useful for generating straightforward and repetitive content, such as financial reports or weather forecasts, where the data is highly structured, and the output needs to follow a specific format. Using NLG in medical reporting can improve the efficiency and accuracy of generating reports, reduce the workload of medical professionals, and enhance patient care. Medical reports are critical documents that contain sensitive information about patients, and accuracy is essential in this field. Therefore, you should ensure that your NLG software is reliable, accurate, and compliant with industry regulations.

Standardized Medical Reports: Rule-based NLG could be used to generate standardized medical reports, such as discharge summaries, progress notes, or pathology reports. These reports often follow a specific format and contain standard language, making them suitable for rule-based NLG.

Patient Education Materials: Patient education is an important aspect of healthcare, but creating personalized educational materials for each patient can be challenging. Rule-based NLG could be used to generate patient education materials based on patient data, such as diagnosis and treatment plans, to provide patients with customized information and resources.

The report could be generated automatically based on patient data, such as electronic health records (EHRs) or clinical notes. The report could also be customized to the patient's language preference and health literacy level, ensuring that the patient can understand and act on the information provided. Providing patients with a customized medical report could help improve patient engagement, satisfaction, and outcomes, while also reducing the workload of healthcare professionals.

Tech Stack

The prototype for our product **EzyMed**, consists of some of the basic functions of our product and demonstrates the actual working of the product.

Dataset

We have used the MIMIC-III from Physionet.org for the prototype. However, the backed is designed to integrate other datasets.

Front-end

We've used HTML, CSS, Bootstrap 3(CSS framework), JavaScript, jQuery v3 for the web application.

Backend

The backend is designed in PHP, (Laravel 8 – PHP framework) & the database is designed in MySQL.

NLG

We've used Arria Studio's APIs to integrate it with our backend.

Workflow (prototype only):

Data pre-processing is done at the backend.

Data and lists are displayed at the front end.

Whenever a report is generated, the backend calls the Arria APIs with the sufficient data in JSON format. The API returns with the generated text. Generated text is displayed in the front-end.

The generation of the medical report is done in Arria Studio, where the scripts are divided according to their functions and the data it is feeded. The basic condition used to form sentences are if clauses to cycle through the lists and most medical conditions. The report itself is divided into 3 parts, the general patient information, the lab and microbiological test prescribed.

The script for general patient info displays the basic patient's information like patient's Id, Date of birth, admit date and time, gender and hospital Id. It uses the variables created to generate sentences. For lab tests, the process is handled by the LabEvents scripts, where it displays the test results and gives general details about the prescribed test and its recommended values for reference. Finally, the Microbiology script works on producing sentences for displaying the test results and gives some details about it for better understanding. The script also makes use of the list function to make the sentences more concise and not repeat the given result.