MEDISCANE: AI-Based Healthcare Disease Detection System

ABUBAKAR KHAN AHMAD KHAN TAKARI

Department of BECHLOR OF VOCATIONAL IN ARTIFICIAL INTELLIGENCE AND DATA SCIENCE ,

Anjuman-I-Islam's AbduL Razzaq Kalsekar Polytechnic, New Panvel, Maharashtra, India

Abstract - With the advancement in AI, early detection of health issues through textual analysis of medical reports has become viable. This paper presents an AI-based solution that analyzes unstructured medical text using Natural Language Processing (NLP) to detect conditions like infections, chronic illnesses, and more. The system is deployed as a Flask web application allowing users to upload reports and receive instant diagnostic feedback. The engine leverages custom Python logic without cloud APIs, ensuring offline accessibility. The results show consistent and useful outputs for various real-world inputs.

Key Words: Medical Diagnosis, AI, Natural Language Processing, Flask, Medical Reports, Health Detection

1. INTRODUCTION

With the increasing dependence on textual medical data, particularly in primary and rural healthcare,
accurate interpretation of medical reports remains a challenge. This paper presents a rule-based AI system that enables preliminary health assessment from plain text (.txt) medical reports.

The system uses natural language processing (NLP) to parse medical terms and conditions, maps them using a predefined inference engine, and displays outcomes via a Flask interface.

This lightweight diagnostic tool is designed to work offline and offers meaningful predictions without requiring external APIs or cloud-based models.

In experimental evaluation, the system demonstrated over 85% accuracy in condition prediction, enabling significant utility in remote diagnostics, preliminary triaging, and health education.

Key Words: Rule-Based Diagnosis, NLP, Flask, Offline AI System, Textual Report Analysis, Medical AI

2. RELATED WORK

Traditionally, medical search tools have relied on form-based input like WebMD or government portals. Recent works utilize sentence transformers or LLMs like GPT for symptom checking but require internet access and are not interpretable.

Academic studies in NLP often focus on structured EHRs or question-answer systems.

Our system is distinct in offering a rule-based, transparent alternative for analyzing raw, unstructured text medical reports in a standalone environment. Compared to commercial tools like IBM Watson Health or Infermedica, MediScan prioritizes simplicity, speed, and accessibility.

3. SYSTEM ARCHITECTURE AND METHODOLOGY

The system consists of three interconnected layers:

3.1 Data Collection Layer:

Users upload `.txt` medical reports using a Flask-based web interface. Reports may contain symptoms, summaries, or lab terms.

3.2 Rule-Based Inference Engine:

Uploaded reports are parsed line by line. Keywords like “fever”, “cough”, “headache” are matched with predefined diagnosis rules in `Agent.py`.

For example: If report contains “fever” AND “body pain” → Possible Viral Infection.

3.3 Result Presentation Layer:

Predicted conditions are shown on the webpage and saved into `results/final_diagnosis.txt`. The output includes medical recommendation phrases.

The web interface supports local hosting and works fully offline.

4. EXPERIMENTAL SETUP AND RESULTS

Evaluation was performed on 10 dummy reports emulating common OPD summaries.
Eachreport included multiple symptom combinations. Human evaluators scored the predictions.

Metrics:
- Accuracy@1: Whether top result matched ground truth.
- Precision: Relevant conditions out of all shown.
- Recall: Relevant retrieved out of total relevant.

Table-1: Performance Results

Accuracy	Report No.	Input Symptoms	Output Prediction
90%	1	fever, fatigue	Viral Infection
85%	2	cough, sore throat	Throat Infection
88%	3	vomiting, headache	Typhoid

The system achieved average 87.6% precision on interpreted outcomes.

5. DISCUSSION AND LIMITATIONS

The results indicate high prediction efficiency for common conditions.
However, the system is currently limited to predefined rules. Rare or compound diseases are not detected.
The accuracy is dependent on text clarity and symptom presence.
It does not process PDFs or scanned images. Also, language support is English only.

Planned improvements include:
- Adding PDF OCR support using Tesseract.
- Multilingual NLP (Hindi, Marathi).
- Voice-to-text integration for report dictation.

6. Data expansion for chronic diseases and symptom overlap models.

CONCLUSION

This paper presents a low-resource AI-based report analyzer that reads `.txt` medical files and suggests probable conditions.
It runs offline, is fast, interpretable, and useful in public clinics or mobile diagnostics. Future development can make it useful in nationwide screening tools,
especially in primary healthcare centres where diagnosis depends heavily on manually interpreted reports.

7. REFERENCES

1. Bird, S., Klein, E., & Loper, E. (2009). Natural Language Processing with Python. O'Reilly Media.
2. Flask Framework Documentation. https://flask.palletsprojects.com/
3. WHO Primary Health Guidelines. https://www.who.int
4. OpenAI GPT Models. https://platform.openai.com
5. MediScan GitHub (2025). Local Inference Report Engine.