Developing an adaptive diagnosis system for Nigeria

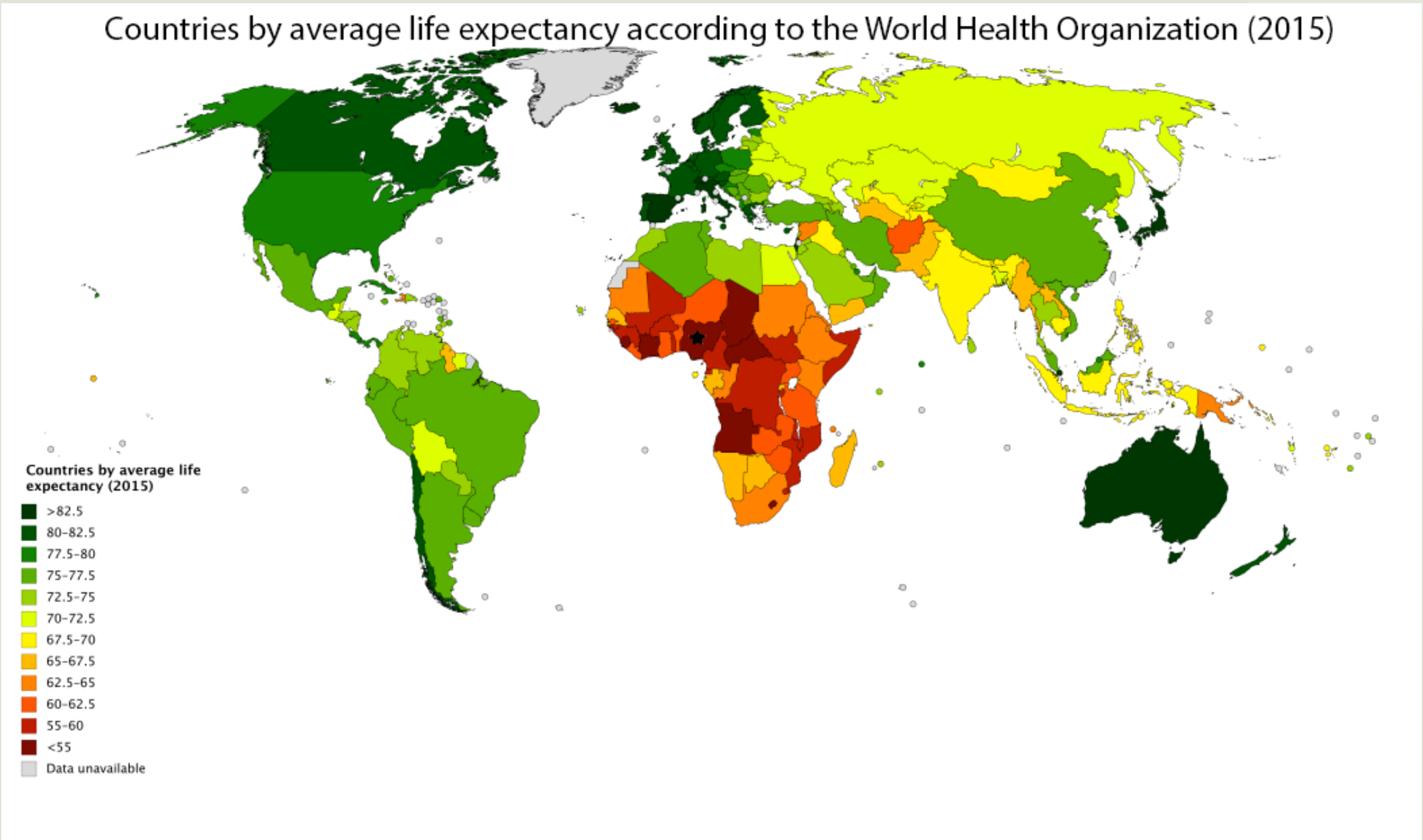
Chao Zhang, Hanxin Zhang, Andrey Rzhetsky

Pritzker School of Molecular Engineering & Institute for Genomics and Systems Biology

Abstract

The clinical decisions strongly rely on doctors' ability to recognize subtle cues and patterns across patient's clinical history and correlate it with the patient's current symptoms. The art and skill of high-fidelity diagnostics takes doctors years to develop. As a result, physicians are more likely to make mistakes when confronted with rare conditions. Diagnostic errors affect an estimated 12 million Americans each year, as from Society to improve diagnosis in medicine. Machines can do much better and faster to learn the patterns. With the help of computers, diagnosis errors has a chance to be minimized.

We are developing a computerized diagnosis system which can help doctors to make more accurate and comprehensive diagnosis and can work for a broad range of diseases. This system will be adaptive with minimum number of questions or tests so that the diagnosis process is quick and easy, and it will leverage several very large data sets (electronic medical records and textbooks) so that it will be more 'experienced' than any clinicians. More importantly the system will learn by itself. The diagnosis results are going to be recorded and used to improve the performance, especially in different regions. We hope to commercialize the diagnosis system and benefit African people as well as other parts of the world.


Countries by average life expectancy according to the World Health Organization (2015)

Objectives

The aim of our system is to help physicians to make diagnosis faster and more accurately. Based on large database, adaptive diagnosis, and reinforcement learning, the system will provide the physicians with accurate and comprehensive suggestions on the patients possible diseases and corresponding tests and treatments.

We work on large electronic medical record data sets. Some statistical results will be summarized and potentially specific patterns will be found to help increase the accuracy of disease diagnosis. This will benefit the field of medication and data science. In the mean time, a smart phone app will be developed based on the diagnosis system, which contributes to both clinical and software development industry.

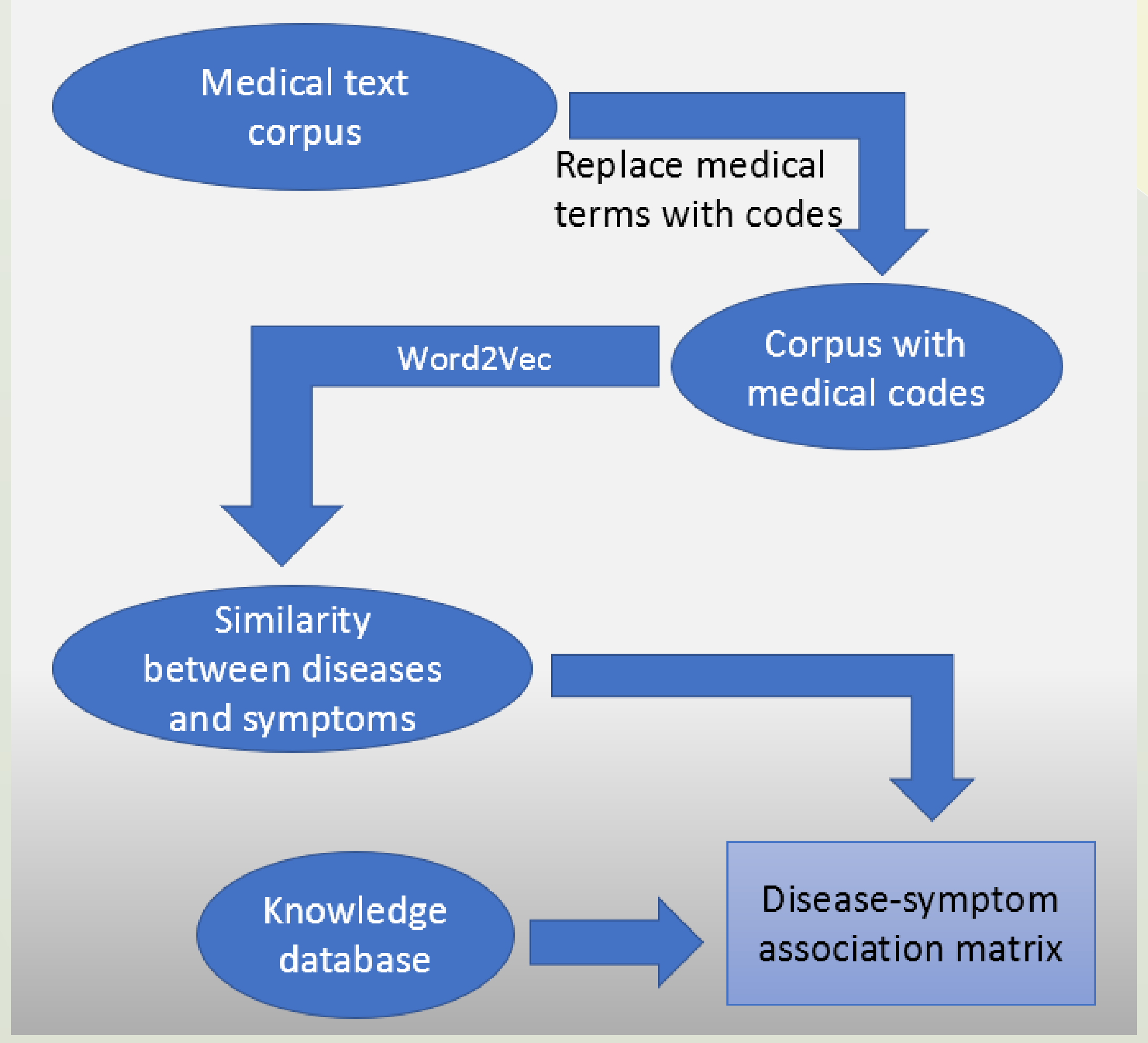
The system is designed to be utilized in Nigeria first and then potentially expand to other parts of the world. Africa is suffering from the lack of clinicians, medical equipment as well as electronic medical record systems. Our mobile app will be able to help increase the accuracy and efficiency of disease diagnosis, and build a standardized recording system, and thus save millions of lives.

Materials and Methods

The current data mainly come from four sources: (1) MarketScan datasets from IBM Watson Healthcare; (2) DeepDive Open Datasets from Stanford University; (3) UMLS from NIH/NLM; and (4) a knowledge database of disease-symptom associations generated with data from New York Presbyterian Hospital admitted during 2004 [24]. We are also exploring for other useful data sets such as clinical notes at the University of Chicago and disease statistics collected in Nigeria.

To link diverse, irregular natural-language phrases to curated/normalized terminology and to quantify the association between diseases and symptoms, we use natural language processing tools, such as Word2Vec proposed by Thomas Mikolov.

Word2Vec models are at their heart simple/shallow neural networks, implemented in several Python natural language processing libraries, such as GenSim


Medical text corpus
Replace medical terms with codes
Corpus with medical codes
Word2Vec
Similarity between diseases and symptoms
Knowledge database
Disease-symptom association matrix

Adaptive diagnosis:

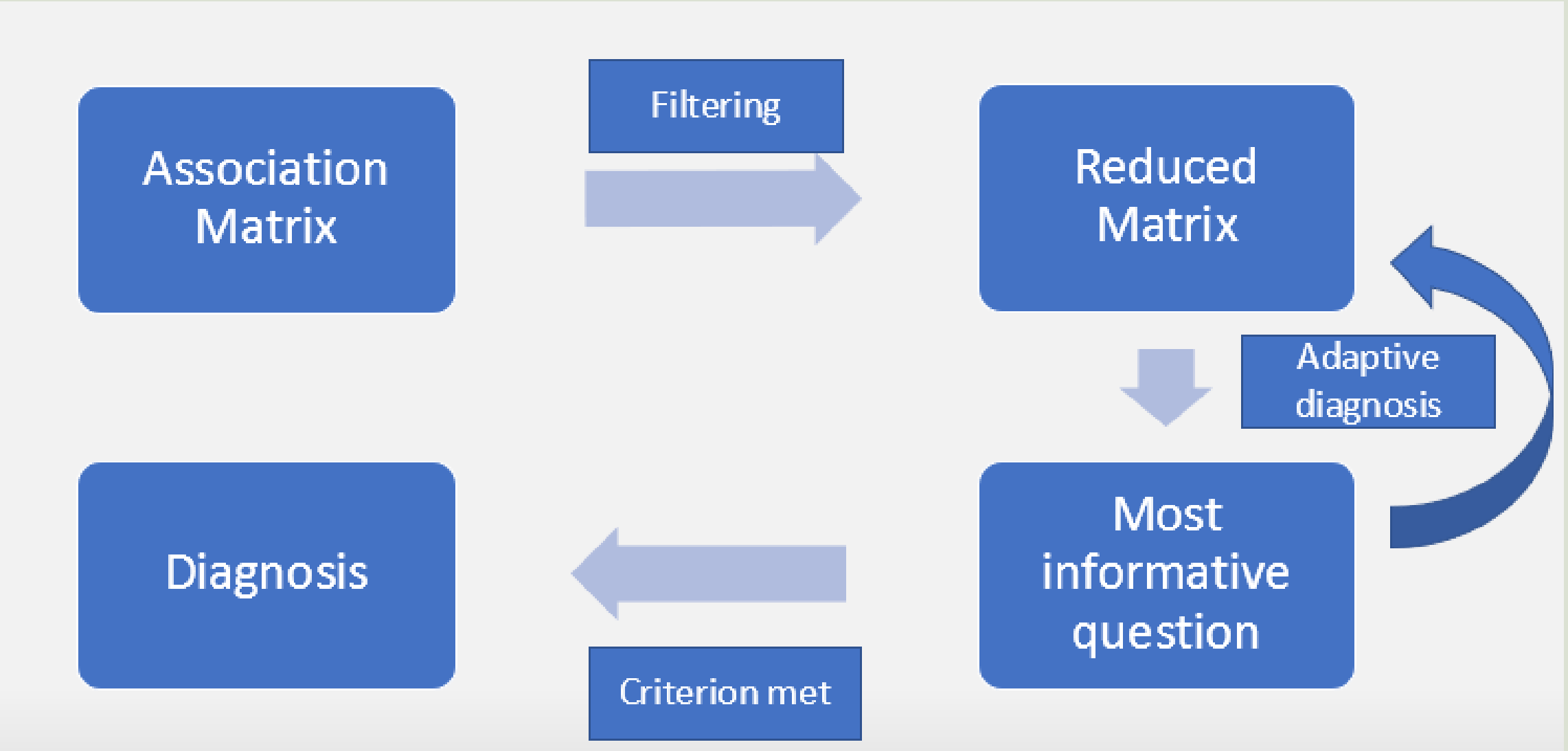
Choose a symptom to ask based on previous answers, to minimize the number to questions.

Questions are selected to minimize the entropy:

$$Entropy = \sum_{i=1}^{C} -p_i \cdot log_2(p_i)$$

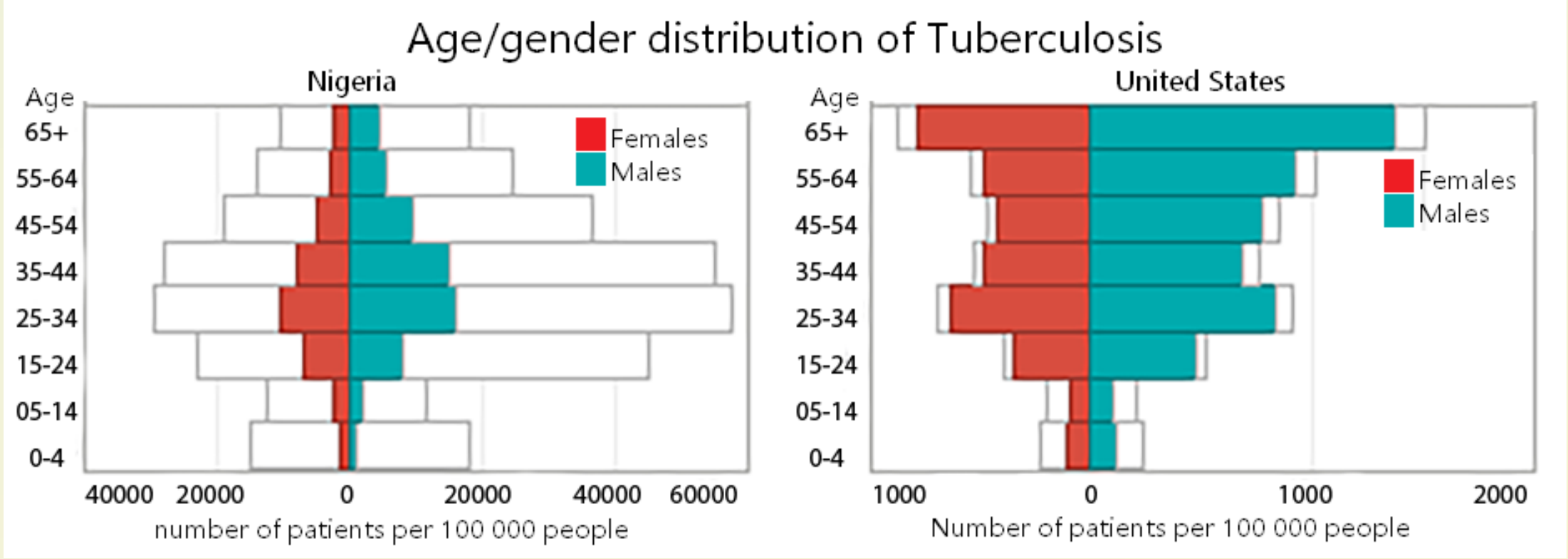
C: number of classes

p_i: probability of class i.

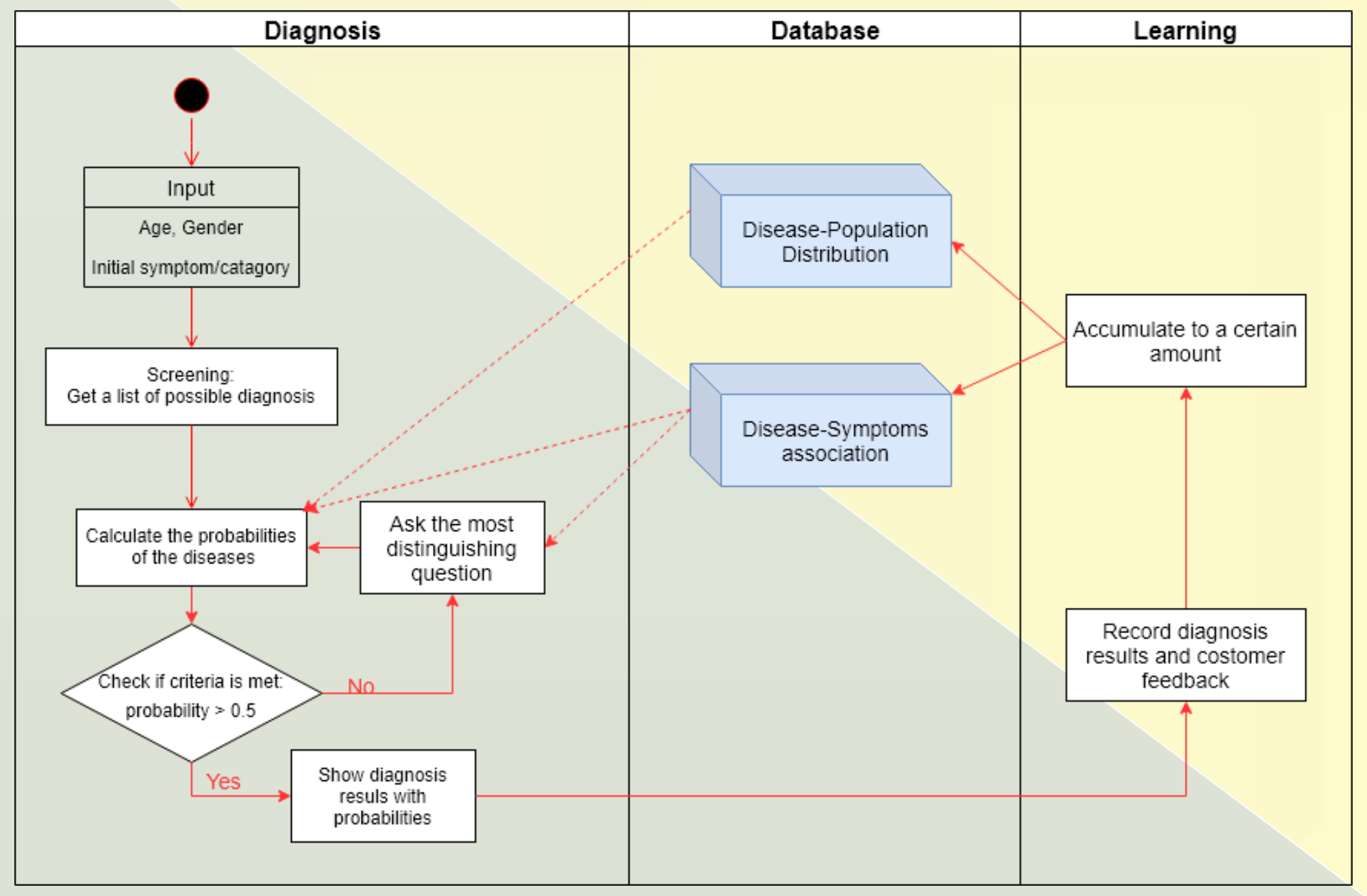


Results

We are building an AI diagnosis system that is explicitly intended for minimizing physician's burden while maximizing utility for patient. Only simple questions will be asked (or tests suggested) and the diagnosis would be typically made in minutes. As shown below, the diagnosis process is consist of input, screening, probability computation, series of questions/tests, and output.


Age/gender distribution of Tuberculosis

Given a limited internet access in many places in Africa, our system will be designed to be able to work offline or via cellular networks (text messages). The diagnostic/statistical dataset will have copies on both server and application side, and regularly updated, once the central server will accumulate enough new information. The diagnostic tool will be design to work locally, on a smart phone. Larger data exchanges between servers and client application of smart phone will happen whenever faster internet access becomes available.


Diagnosis
Database
Learning
Input
Age, Gender
Initial symptom/catagory
Screening:
Get a list of possible diagnosis
Calculate the probabilities of the diseases
Ask the most distinguishing question
Check if criteria is met:
probability > 0.5
No
Yes
Show diagnosis resuls with probabilities
Disease-Population Distribution
Disease-Symptoms association
Accumulate to a certain amount
Record diagnosis results and costomer feedback

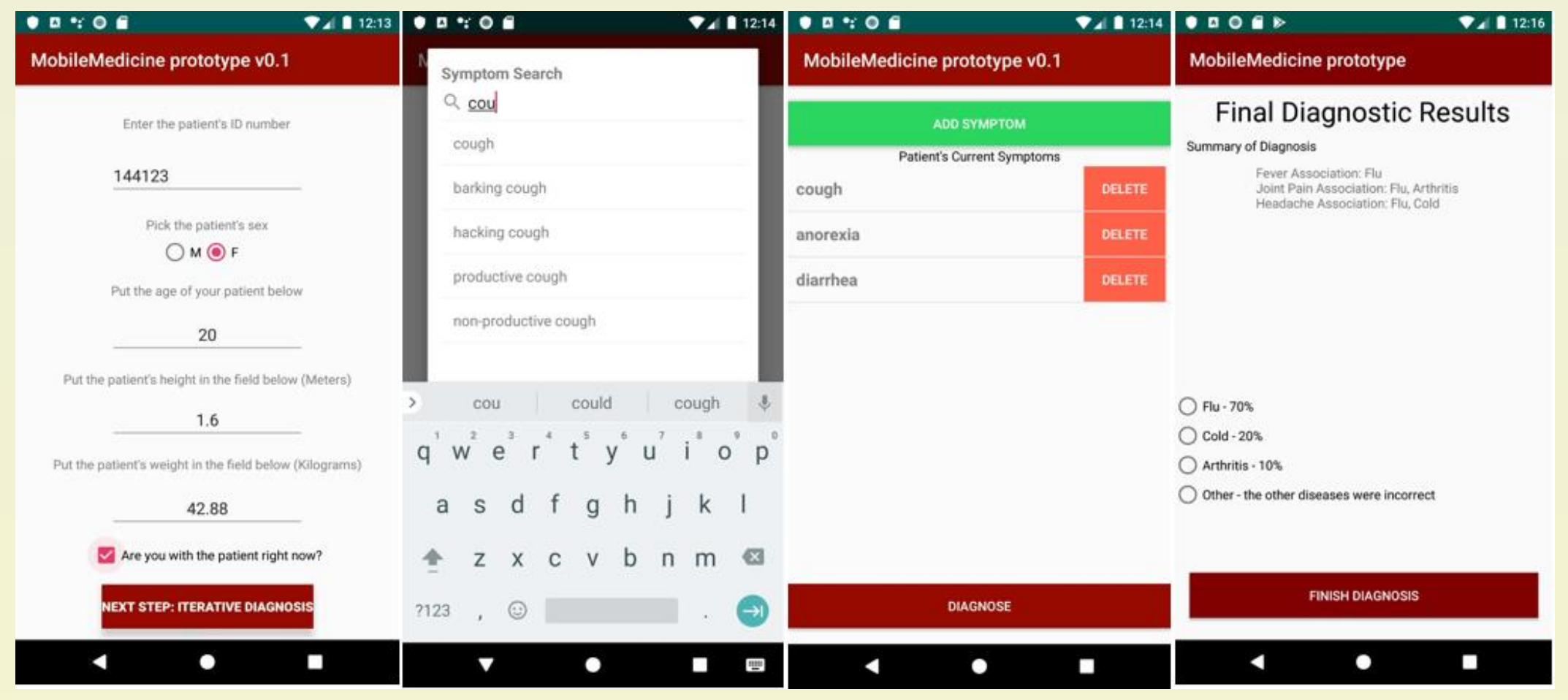
The first step for a physician using our system will be to obtain the basic information from the patient. Based on the statistical distribution of diseases over large populations, a prior hypothesis for the possibilities of diseases for this particular patient are gathered from the general distribution of diseases given sex and age.

The second step performed by the system will involve prioritizing candidate diagnoses given the symptoms the patient has. First, patient complaints and apparent symptoms will be recorded and transformed into unified/normalized medical codes. The most likely diagnoses will be ranked by probability and presented to the physician for evaluation and further analysis.

After computing a refined list of candidate diseases (diagnoses), the doctor will be presented with a possibility to ask the patient a series of clarifying question to reduce uncertainty of the diagnosis in the smallest number of steps.

Conclusion

A prototype of our diagnostic system developed for an Android smart phone.


12:13
MobileMedicine prototype v0.1
Enter the patient's ID number
144123
Pick the patient's sex
○ M ● F
Put the age of your patient below
20
Put the patient's height in the field below (Meters)
1.6
Put the patient's weight in the field below (Kilograms)
42.88
☑ Are you with the patient right now?
NEXT STEP: ITERATIVE DIAGNOSIS
12:14
Symptom Search
cou
cough
barking cough
hacking cough
productive cough
non-productive cough
cou
could
cough
12:14
MobileMedicine prototype v0.1
ADD SYMPTOM
Patient's Current Symptoms
cough
DELETE
anorexia
DELETE
diarrhea
DELETE
DIAGNOSE
12:16
MobileMedicine prototype
Final Diagnostic Results
Summary of Diagnosis
Fever Association: Flu
Joint Pain Association: Flu, Arthritis
Headache Association: Flu, Cold
○ Flu - 70%
○ Cold - 20%
○ Arthritis - 10%
○ Other - the other diseases were incorrect
FINISH DIAGNOSIS

The database is generated by an automated method based on information in textual discharge summaries of patients at New York Presbyterian Hospital admitted during 2004 [24].

Provided with standardized UMLS codes, the knowledge database contains a population of diseases as well as binary associations between diseases and unified/normalized symptoms.

Future plans and collaborations

Phase 1: Prototype building and integration with the diagnosis system, three months;

Phase 2: Testing in Nigeria with a small group of physicians, six months;

Phase 3: Refinement and preparation for large-scale testing, three months

Acknowledgement

The Lab website:
http://www.igsb.org/labs/andrey-rzhetsky

Thanks to University of Chicago Research Computing Center for the computing resources and support.