Automated Emergency Room Triage: Helping Patients Get the Best Treatment

Alexandre Hoppe Inoue, Marcus Vinicius de Alvarenga Prado, Fábio G. Cozman

[1]Escola Politécnica – Universidade de São Paulo (USP) São Paulo – SP – Brazil

alexandre.hoppe.inoue@usp.br, marcus.prado17@gmail.com, fgcozman@usp.br

Abstract. *We describe an intelligent triage system for emergency rooms; the system interacts with patients and classifies them by priority level and with respect to medical specialty. The system consists of a conversational interface, coupled with sensors and a physical robot-like platform, and classifiers that operate on symptoms and measurements so as to select a medical specialty and to output a priority level. Tests with human subjects demonstrated that our Healthbot system was well received and in fact preferred to alternatives. Tests have also shown that the classifiers reached accuracy consistent with a doctor's output.*

1. Introduction

Emergency rooms in Brazil suffer from long waiting lines — one can even find an application that informs the waiting time in each hospital.[1] This is due to a high number of waiting patients per nurses involved in care. A significant portion of a nurse's time is spent in the triage of patients; simple cases take too much time out of their schedule. A successful automated triage system would allow nurses to devote more time to patients with severe conditions. It is thus natural to look for automated triage devices that have conversational agents (chatbots) coupled with decision making by classifiers.

The goal of this work was to develop a complete automated emergency room triage system that combines a chatbot, a robotic interface, a system for automated measurement collection, and classifiers to select a medical specialty and a priority level for a given patient. The Healthbot system described here focuses on the patients with less urgent demands. The system is based on hospital protocols and uses the same kind of information that decision makers involved in the screening process use, performing triage while maintaining a friendly demeanor, with duration compatible with those performed in hospitals.

The Healthbot system was developed in collaboration with a medical research team, and we hope this work contributes with some useful ideas in the quest for useful automated help for patients. The system has some unique features as compared to alternatives, in particular offering a complete solution and supporting Portuguese as a primary language.

Emergency room triage must follow a strict protocol, usually based on the Manchester Triage System protocol also known as MTS [Mackway-Jones 2014] as this latter

[1] R. Okumura, *Aplicativo conecta paciente a pronto-socorro com menos filas*, available at https://saude.estadao.com.br/noticias/geral,aplicativo-conecta-paciente-a-pronto-socorro-com-menos-filas.

protocol is widely regarded as relatively effective [Azeredo 2015]. Given that many hospital protocols directly follow or are based on MTS, we have based our system on similar guidelines, and we have built classifiers that operate on the kind of information generated through MTS — however, again we note that our efforts are particularly relevant for the Brazilian setting as we benefited from the shared expertise of Brazilian doctors.

The paper is organized as follows. In Section 2 we present the main ingredients of our problem and our solution. Section 3 reports on results with real users, analzying both the accuracy of the complete system and the opinion of users about their interactions. Final comments can be found in Section 4.

2. Emergency room triage: a solution

Our automated solution to patient screening consists of three elements, all depicted as a pipeline in Figure 1.

The first two elements obtain and display information. The first element is patient interaction through dialogue with a chatbot, mediated by a friendly physical interface. The second element is the capture and measurement of vital signs such as temperature, pressure, heartbeat and blood oxygenation.

The third element is a classification system that, based on the interaction with the user and the collected information, assigns a level of risk to the user and determines which medical specialty should her be handed to. Two classifiers were implemented, one for priority (risk) level, and the other for medical speciality selection. While the former is a rule-based classification scheme translated from domain expertise, the second is a data-driven classifier learned by statistical methods.

Together, these elements offer a more complete solution than piecemeal approaches in the literature.

2.1. The Healthbot: architecture and infrastructure

The initial interaction with the chatbot and the measurement of vital signs has no mandatory order. They are both combined during interactions with the patient; these interactions happen through a tablet in a robot-like interface depicted in Figure 2(left). This robot-like interface is a slight modification of an open source project, the *Joy Robot* [Albuquerque, 2016] that was provided to us by an associated research team. The goal of this robot-like interface is to offer a friendly connection with the human user.

With respect to measurements, we used the DS18B20 sensor (Maxim Integrated) for temperature measurement and the smartwatch X9PRO to capture the remaining re-

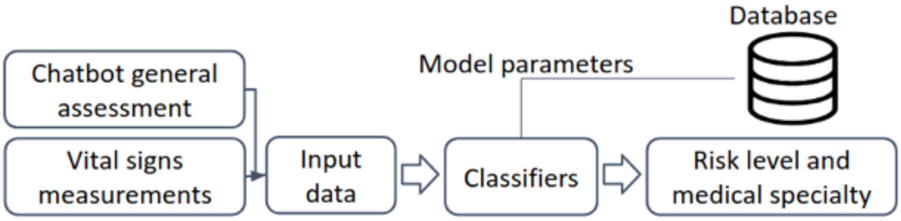


Figure 1. The system pipeline.

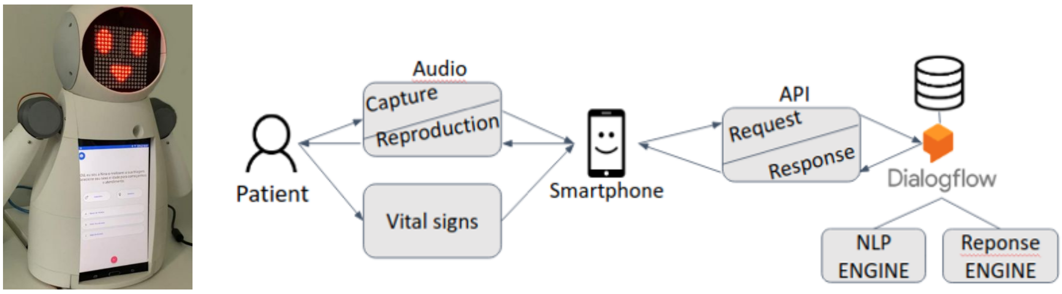


Figure 2. Left: physical user interface with the tablet (smartphone) responsible for processing interactions. Right: architecture of the interfacing system.

quired signals. These devices are easy to use and any patient can thus provide the required data with minor effort. All of these data are transmitted via bluetooth to the tablet.

The tablet captures audio of the user's speech under screening, locally running voice recognition and processing and, via bluetooth communication, receiving data on the patient's vital signs. The data is used by a chatbot that relies on a database of interactions built for the project.

The overall architecture of the interfacing part of the Healthbot is depicted in Figure 2(right). The following modules run in the tablet, each on its own thread (to prevent a loss of execution frames that could harm the interactions):
1. Speech Recognizer Module: Responsible for patient speech recognition and for converting captured audio to text. It uses the Android operating system built-in speech recognition services.
2. DialogFlow Communicator Module: Responsible for the communication with the chatbot.
3. Text To Speech Module: Responsible for playing audio from text.
4. Bluetooth Serial Module: This module communicates with an Arduino processor in the robot-like interface that controls the robot's physical structure: it communicates with the robot's temperature measurement system and it communicates with the vital signs measurement system.
5. Patient Module: This module saves all patient data captured during the interaction and runs the communication with the classifiers' server.

On top of these modules, the running system also has a MessageHandler whose main functions are:
1. Receive and handle the results of parallel module processes.
2. Manage and forward results for presentation to the user.

Our implementation follows an MVC (model-view-controller) design. A main class works as a controller, triggering modules as needed and decided which ones will be available in the view. The view is responsible for the audiovisual communication with the user (represented mainly by XML elements). Modules perform the model role of the application, processing the data or communicating with the servers. As modules terminate their processes, their output reaches the current activity through the MessageHandler, enabling a fluid user experience.

2.2. The chatbot

The chatbot collects patient utterances so as to understand the symptoms and their intensity, as well as to capture the overall state of the patient. The chatbot also collects key information such as the patient's possible drug allergies.

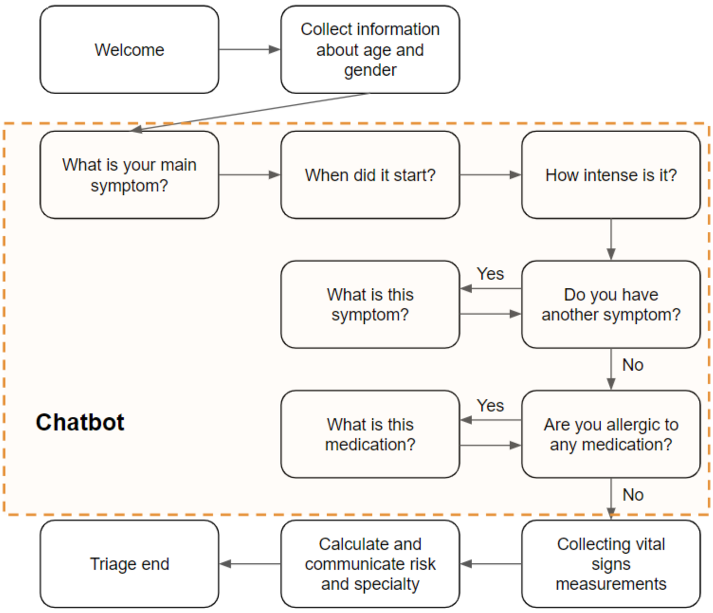

Welcome
Collect information about age and gender
What is your main symptom?
When did it start?
How intense is it?
What is this symptom?
Yes
Do you have another symptom?
No
Chatbot
What is this medication?
Yes
Are you allergic to any medication?
No
Triage end
Calculate and communicate risk and specialty
Collecting vital signs measurements

Figure 3. State model that describes the service script followed by the chatbot; different states correspond to distinct utterances and exchanges.

Conversational agents (chatbots) are currently used in a number of large enterprise customer service issues [Jurafsky & Martin 2018], and it is not surprising that they have been tested in healthcare. For instance, the company Baidu has launched the Melody Chatbot in China so as to mitigate problems with China's healthcare system. The Melody Chatbot uses deep learning and natural language processing to run a patient's anamnesis, gathering information about the symptom's frequency, intensity and duration. The chatbot helps the patient better understand her clinical condition before consulting a doctor, thus making the interaction with the doctor to be more fruitful and saving time for the physician time. The information collected by the Melody Chatbot is stored in a patient's story that is made available to doctors. Similar systems are *Your.MD*,[2] *Sensely*,[3] and *Ada Health*,[4] as all of them work through dialogue with a chatbot that consults information in available databases. Ada Health even promises to evolve to a predictor of patient health.

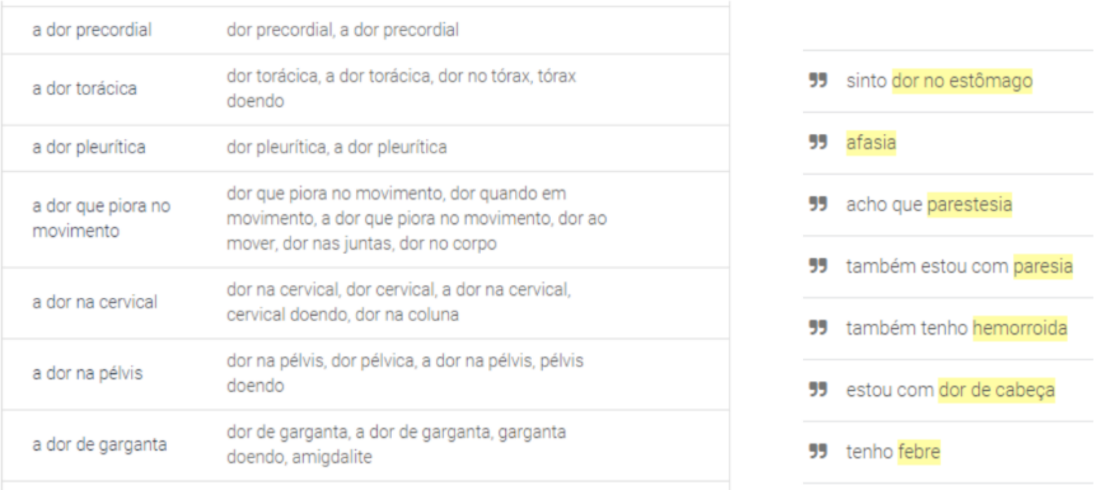
Our chatbot is based on a state model. This model was built with help from an expert doctor and through conversation with hospital officials so as to capture the triage process; the state model is depicted in Figure 3.

As a patient may describe symptoms in several different ways, the Healthbot should be able to link a patient utterances with the correct symptom. Besides, the Healthbot should recognize as many symptoms as possible. With such goals in mind, it was

[2] Your.MD Health Guide and Symptom Checker, available at (2019): https://www.your.md/

[3] Sensely – How are you feeling today? available at (2019): http://www.sensely.com/t/.

[4] Ada: Your Personal Health Guide, available at (2019): https://ada.com/.



a dor precordial	dor precordial, a dor precordial
a dor torácica	dor torácica, a dor torácica, dor no tórax, tórax doendo
a dor pleurítica	dor pleurítica, a dor pleurítica
a dor que piora no movimento	dor que piora no movimento, dor quando em movimento, a dor que piora no movimento, dor ao mover, dor nas juntas, dor no corpo
a dor na cervical	dor na cervical, dor cervical, a dor na cervical, cervical doendo, dor na coluna
a dor na pélvis	dor na pélvis, dor pélvica, a dor na pélvis, pélvis doendo
a dor de garganta	dor de garganta, a dor de garganta, garganta doendo, amigdalite

sinto dor no estômago

afasia

acho que parestesia

também estou com paresia

também tenho hemorroida

estou com dor de cabeça

tenho febre

Figure 4. Left: examples of some symptoms and its synonyms in Portuguese. Right: examples of training phrases in Portuguese.

necessary to build a database of symptoms and their synonyms (Figure 4 (left)), when building the synonyms database we had the goal of covering the different ways of referring to the symptom, from the most formal (original symptom) to the most informal. Note that our system operates in Portuguese, so the words and expressions in the database are in that language.

When the chatbot receives a message it has to detect the entities in it; we trained the chatbot by giving examples of sentences that we expect the user to say. We tried to cover most of them by giving these training phrases as examples on DialogFlow (Figure 4 (right)).

One particularly novel characteristic of the chatbot is that it looks for possible symptoms that the patient may have, using an automatically built graph of symptom similarity. The similarity between symptoms is used by the chatbot in the state where it must ask about other symptoms; whenever a close symptom is found, the chatbot asks whether the patient is feeling that particular symptom. The similarity between two symptoms is computed using the database of symptoms (described later in connection with classifiers); from that database we compute the cosine similarity between symptoms.

The whole graph of symptom similarity is depicted in Figure 5. Each symptom is a node colored according to the most frequently observed medical specialty for that symptom; the graph was drawn using the Fruchterman-Reingold algorithm by taking similarities as force-weights (with aid of NetworkX python package). One can see in Figure 5 not only the strong correlation between symptoms and medical specialties, but also the prevalence of the "general practitioner" class in referrals.

On the implementation side, as chatbot development framework we have chosen the DialogFlow package due to its support for Portuguese (Brazilian). We worked specifically on three aspects of the chatbot:

- **Entities:** These are the objects that we want to know about from the received sentences, the main one being the symptom entity that connects what the patient said with any of our base symptoms. Numeric and date entities were used as

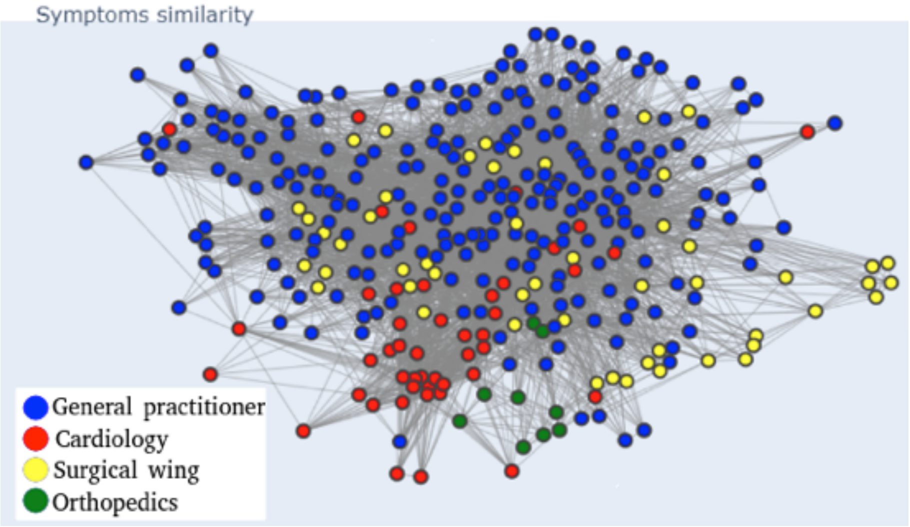
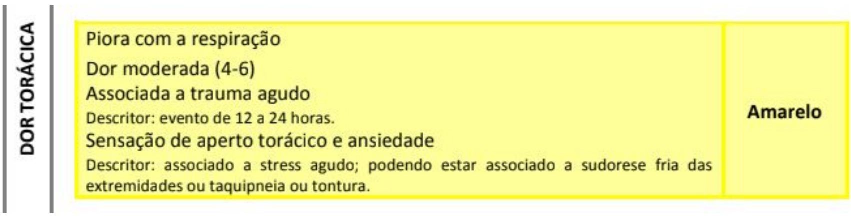


Figure 5. Symptom similarity graph.



DOR TORÁCICA	Piora com a respiração Dor moderada (4-6) Associada a trauma agudo Descritor: evento de 12 a 24 horas. Sensação de aperto torácico e ansiedade Descritor: associado a stress agudo; podendo estar associado a sudorese fria das extremidades ou taquipneia ou tontura.	Amarelo

Figure 6. Risk classification prescribed by Risk Rating Handbook.

supported by DialogFlow.

- **Intents:** These are basically the states of the conversational agent;they were developed so that hospital's service protocols are completely followed.
- **Contexts:** The objects that are responsible for connecting intentions, when leaving an intention the context of the next intention of the proposed sequence is activated.

The DialogFlow package includes a natural language processor, the NLP Engine, that parses the sentences by the user. Once symptoms, vital signs, patient pain level and so on are identified, the Response Engine is in charge to continue the interaction with the patient. This response is sent to the tablet, that plays it to the patient.

2.3. Risk level classifier

To develop the classifier of risk level (that is, priority level for the patient), we followed risk classification procedures that are adopted by hospitals — such procedures are typically approved by health departments and applied by nurses in emergency rooms.

The risk classification protocol we adopted is described in the Brazilian Federal District Health Secretariat's Reception and Risk Rating Handbook [Ministério Da Saúde 2018]. This handbook contains textual classification trees as depicted in Figure 6 for thoracic pain.

The handbook prescribes five risk classes identified by colors (note that Figure 6 depicts a classification rule for the yellow class):

Red: Patient at imminent risk of death and in need of immediate care.

Orange: Patient at risk of worsening condition and in need of continuous assistance (target time for the care of those patients is up to 10 minutes).

Yellow: Patients who can be treated in a first-come, first-served basis and that can receive preventive measures at any time (target time for the care of those patients is up to 60 minutes).

Green: Patients without risk of injury that can be seen on a first-come, first-served basis (target time for the care of those patients is up to 6 hours).

Blue: Patients at the lowest risk level considered, treated on a first-come, first-served basis (target time for the care of those patients is up to 12 hours).

All rules in the handbook (for more than a hundred cases) were translated into a rule-based classifier for risk level. The classifier works as indicated in the handbook: it fits a patient in a certain case based on its main symptom; then it tries to fit a patient in a risk classification starting from the red risk and ending on the blue risk. Each color classification is connected to rules based on symptoms, pain/discomfort intensity and vital signs measurements.

2.4. Medical specialty classifier

The development of the medical specialty classifier was based primarily on a dataset in the literature[5] that was generated automatically through analysis of textual discharge summaries of patients at the New York Presbyterian Hospital [Wang et al. 2008]. The database consists of disease-to-symptom associations, also including the frequency rate of how many patients appeared with their disease in the hospital.

As our goal is to classify the medical specialty rather than the patient illness, an expert doctor assisted us in expanding the original database by mapping out which medical specialty should treat each disease within the Brazilian scenario. Additionally, the number of observations in the original database, using the frequency rate feature of each disease, was expanded so that each row of the database represented a patient who should be referred to the expert-defined medical specialty. This was done using a “dropout” technique: each generated patient receives a random subset of all symptoms characteristics of his illness. This increases variability and makes the data more representative of actual scenarios. For instance, patients do not necessarily go to an emergency room with all the characteristic symptoms of an illness; even when they do that, they might not report less bothersome symptoms during the triage. Furthermore, in some cases, during the triage a patient may intentionally omit symptoms. The classifier has to have some resilience to all those phenomena.

Before expanding the database though, one must attend to the fact that the dataset is highly imbalanced towards more common medical specialties, as can be seen in Figure 7. This dataset imbalance was remedied through patient random oversampling of the less common classes.

The classifier was trained by Catboost, an ensemble procedure that resorts to boosting (we employed an open-source implementation by Yandex). The dataset was divided into training and validation in a stratified manner, so as to maintain the class pro-

[5] Available at: http://people.dbmi.columbia.edu/ friedma/Projects/DiseaseSymptomKB/index.html.

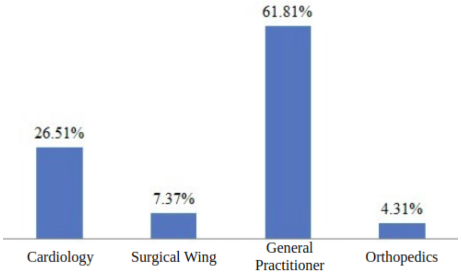


Figure 7. Medical specialty class imbalance.

portions in each dataset. The final classifier was an ensemble of 1314 trees obtained after training with 100 iterations of early stopping.

3. Evaluation of the Healthbot

To evaluate our system we tested patient communication with the Healthbot, patient satisfaction with provided response and patient classification accuracy both for risk level and for medical specialty.

3.1. Test structure

We evaluated, first, whether the chatbot can understand what the patient stated, and whether it can correctly understand the symptoms declared by a patient. For the second part of the test, patient satisfaction, the goal was to verify whether our resulting triage is comparable to a real one. For this we measured the time spent in each triage simulation, the success triage conclusion ratio, whether the patient would prefer the chatbot triage or one with a real nurse, and a grade for both satisfaction and robot appearance.

The tests were performed by 40 people as a pooling exercise (so that ethical rules were satisfied). All participants were apparently healthy; we asked them to use the system and to simulate some illness, referring to the last time they've been to an emergency room. Prior to the start of the test, an explanation of the project was given and the participant was asked about what he was feeling the last time he went to the emergency room. The user then interacted with the Healthbot; at the end of the triage, the patient answered a questionnaire.

The third part of the test aimed to measure the quality of ours classifiers; for that a dataset based on disease symptoms was created,[6] and related symptoms were also used by resorting to our graph of symptom similarity (Figure 3). To increase the diversity of cases in the dataset, we varied the intensity of the symptoms in each of the examples. We also took examples for each age group considered in the models (pediatrics, adults and the elderly). With this database in hand, an expert doctor classified this entries with a risk level and a medical specialty.

The risk level classifier classifies the patient in 5 different risks; for our evaluation we divided the possible comparisons to the doctor classifications in accurate (when our risk level is the same as the doctor's), slight mistake (when our risk level is higher), critical mistake (when our risk level is lower).

[6] Data from Sabará - Hospital Infantil, available at https://www.hospitalinfantilsabara.org.br/categoria-sintomas-doencas-tratamento.

For the medical specialty classifier, general classifier metrics were used: total accuracy; accuracy for each class; precision, recall and F1-score.

3.2. Results

By applying the previously defined test structure, the proposed intelligent triage system yield results summarized in Figures 8 and 9.

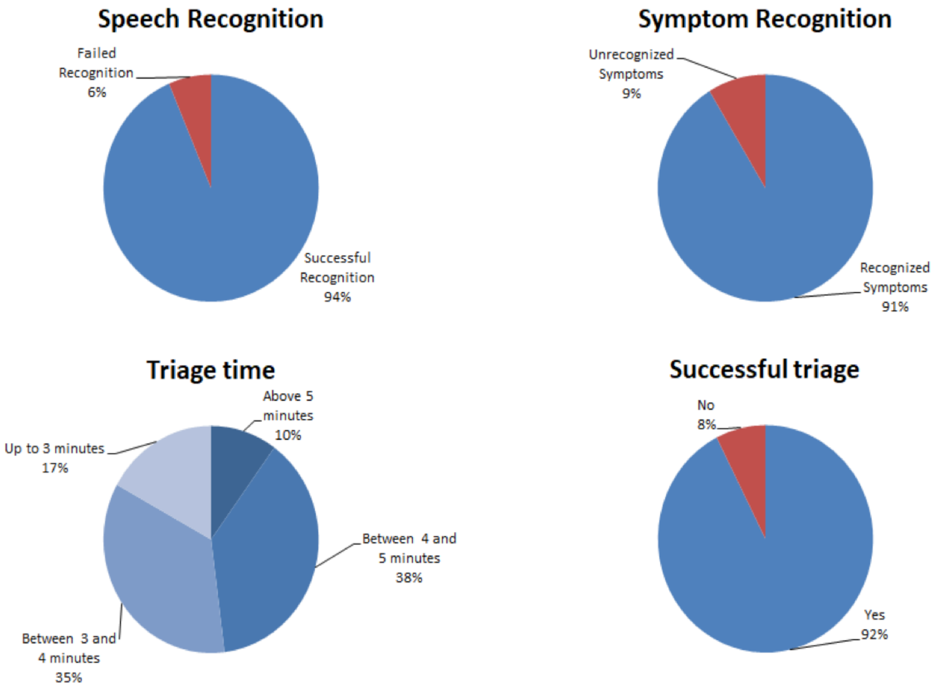


Figure 8. Results of user tests.

Figure 8 presents the results for the first and second part of our tests with users, while Figure 9 shows our classifiers performance tested in the created separate database.

- Average triage time: 3 minutes and 55 seconds;
- Satisfaction score average: 8,725;
- Average Appearance Rating: 9.425.

3.3. Discussion

Most of the tests were performed at the Escola Politécnica of the Universidade de São Paulo with students and faculty members; other tests were performed in other locations within the Universidade de São Paulo.

It should be noted that the participants are overall in daily contact with new technologies and have a higher level of education; other biases can be seen in the graphs of Figure 10 as they indicate that most of our participants are male and are between 18 and 65 years old. Ideally more tests should be done in the future with a larger number of people with real illnesses, in first aid and with appropriate measuring equipment.

The average triage time was 3min55s, tests showed that most of this time was spent taking the temperature measurement, which took approximately 1 minute and 30

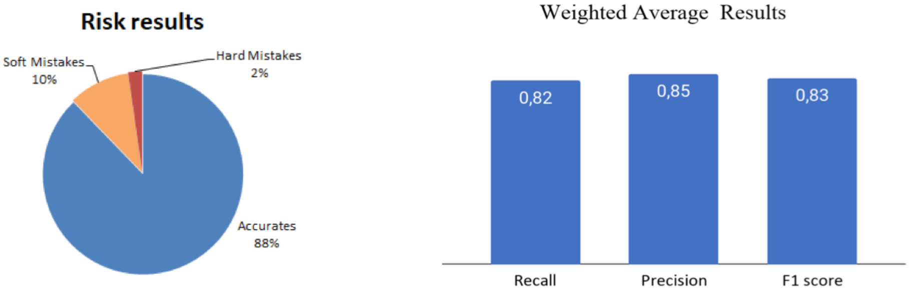


Figure 9. Results of risk (left) and specialty (right) classification.

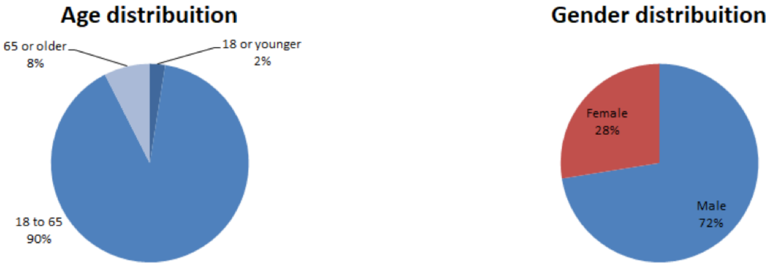


Figure 10. Age and gender distribution in user tests.

seconds. It would be possible to reduce this time by using more advanced measuring equipment. As analyzed by Chianca (2016), the average screening time done by a nurse is 2 minutes and 6 seconds, which is less than the average screening time performed by Healthbot but approximately the same time excluding the measurement section. In any case, the average waiting time for screening to begin with a nurse is 12min33s and this can be significantly reduced given availability of the Healthbot.

Concerning grades, both were high, especially appearance's (average 9,425). The physical robot was attractive to people. While the satisfaction score encompasses more elements, although good (average 8,725), it can be improved by improving the measurement system and design.

Given that selection of medical specialty is a problem with 4 different classifications, its classifier displayed excellent accuracy. The global accuracy was 82.2% and even each class accuracy is at a high value, between 79% and 90%. However, when one analyzes each class result of precision, recall or F1-score, although they are consistent, some room for improvement emerges. On one hand, classes such as cardiology and general practitioner showed balanced results and high recall and precision scores, resulting in a good F1-score as well. On the other hand the orthopedics and surgical wing classes showed high recall values but low precision score. In other words it has a high rate of true positives but it also has a high rate of false positives. These surgical wing results can probably be explained by the symptom similarity graph (Figure 5): a large part of the surgical wing symptoms are separated on the right side of the graph, however there are many more of these symptoms mixed with symptoms of other classes in the middle part of the graph. A similar problem is that the orthopedics class has its main symptoms well separated from the other classes, but some of the symptoms that are characteristic of this class are more frequent in the samples of other classes, mainly because the class is less prevalent in the dataset.

4. Conclusion

The ultimate goal of the project is the development of an intelligent system capable of performing first aid screening so that the automated triage process should be comparable to those performed by practitioners dealing with less urgent cases. We have developed the Healthbot with this goal in mind and in the context of Brazilian hospitals — however we feel that some of its best ideas can be applied to other locations.

The user interface of the Healthbot has been deemed attractive and efficient; despite some classification errors, most users enjoyed the experience of being serviced by the system and would like to have this option of first aid. The user tests were generally very positive and served as an incentive for future work. The classifiers, which are key elements of the project, achieved good accuracy even with simple models and relatively small datasets. The classifiers seem to perform close to nurses, but more diverse tests with a larger number of participants would be needed to confirm this statement. Overall, the triage system presented in this paper can be contemplated in real settings; future work should be directed to refining it and extensively testing it in practice.

Given the fact that our risk model was based on a handbook that is used in hospitals, it displayed rather good accuracy when compared to a doctor's classifications. However, some problems were found in tests, and future work should be directed to fixing them. The main problem is that our system does not measure everything that is required by the hospital handbooks; for example, the respiratory frequency is not measured (however, when a doctor sees some symptoms related to respiratory problems, she can accurately estimate a respiratory level and give a better classification). Another problem is that the model is based on a static handbook and does not learn from past cases. A possible improvement would be a machine learning model that trains on a historical database and that could learn over time as the database gets larger. Classifiers should be adapted to lifelong learning when used in hospitals, as new symptoms may emerge and risk and specialty ratings may vary over time. Even though our proposed solution deal only with patients with less urgent demands, a discussion of the ethical implications of this technology as well as further tests involving a larger and more representative groups of patients are necessary before deployment in a real hospital. We hope this paper opens a debate around such questions.

Acknowledgements

The authors are grateful to Doctor Miguel Moretti for his medical advice. We also thank Helder Nakaya and Igor Albuquerque for providing the robot interface; Mariana Fraga for software support; Luiz Yamaoka for technical insights.

The third author has been partially supported by the Conselho Nacional de Desenvolvimento Cientifico e Tecnológico (CNPq), grant 312180/2018-7. The work was also supported by the Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), grants 2016/18841-0 and 2019/07665-4, and also by the Coordenação de Aperfeiçoamento de Pessoal de Nivel Superior (CAPES) - finance code 001.

5. References

Albuquerque, I. (2016) Joy Robot (Robô Da Alegria) - Open Source 3D Printed, Arduino Powered Robot! Available at: https://www.instructables.com/id/Joy-Robot-Robo-Da-Alegria-Open-Source-3D-Printed-A/.

T. R. M. Azeredo, H. M. Guedes, R. A. R. de Almeida, T. C. M. Chianca, J. C. A. Martins. Efficacy of the Manchester triage system: a systematic review. *International Emergency Nursing*, 23(2):47–52, 2015.

T. C. M. Chianca, R. M. Costa, M. V. Vidigal, L. C. R. Silva, G. A. Diniz, J. H. V. Araujo, C. C. Souza. Tempos de espera para atendimento usando Sistema de Triagem de Manchester em um hospital de urgência. *REME – Revista Min. Enferm.*, volume 20, 2016

B. E. V. Comendador, B. M. B. Francisco, J. S. Medenilla, S. M. T. Nacion, T. B. E. Serac. Pharmabot: A pediatric generic medicine consultant chatbot. *Journal of Automation and Control Engineering*, 3(2):137–140, 2015.

A. Coucke, A. Saade, A. Ball, T. Bluche, A. Caulier, D, Leroy, C. Doumouro, T. Gisselbrecht, F, Caltagirone, T. Lavril, M. Primet, J, Dureau. *Snips Voice Platform: an embedded Spoken Language Understanding system for private-by-design voice interfaces*, https://arxiv.org/pdf/1805.10190.pdf, 2018.

V. B. E. R. Feijó, L. Cordoni Jr., R. K. T. de Souza, A. O. Dias. Análise da demanda atendida em unidade de urgência com classificação de risco. *Saúde Debate* 39(106):627–636, 2015.

D. Jurafsky, J. H. Martin. *Speech and Language Processing*, 2018.

S. Levin, M. Toerper, E. Hamrock, J. S. Hinson, S. Barnes, H. Gardner, A. Dugas, B. Linton, T. Kirsch, G. Kelen. Machine-learning-based electronic triage more accurately differentiates patients with respect to clinical outcomes compared with the emergency severity index, *Ann Emerg Med.*, 71(5):565–574, 2018.

K. Mackway-Jones, J. Marsden, J. Windle. *Emergency Triage: Manchester Triage Group*, 3rd Edition, 2014.

D. Milward, M. Beveridge. Ontology-based dialogue systems *Workshop on Knowledge and reasoning in practical dialogue systems (IJCAI03)*, 2003.

Ministério da Saúde do Brasil. *Manual de Acolhimento e Classificação de Risco*; 2018.

Y. Raita, T. Goto, M. K. Faridi, D. F. M. Brown, C. A. Camargo Jr, K. Hasegawa. Emergency department triage prediction of clinical outcomes using machine learning models, *Crit. Care*, 23(1):64, 2019.

X. Wang, A. Chused, N. Elhadad, C. Friedman, M. Markatou. Automated knowledge acquisition from clinical reports. *AMIA Annu Symp Proc.*, pp. 783–787. 2008.