Cluster Computing (2020) 23:2471–2482
https://doi.org/10.1007/s10586-019-03020-1

SPeCECA: a smart pervasive chatbot for emergency case assistance based on cloud computing

Nourchène Ouerhani[1,2] · Ahmed Maalel[1,2] · Henda Ben Ghézela[2]

Received: 10 May 2019 / Revised: 30 June 2019 / Accepted: 20 November 2019 / Published online: 28 November 2019

Abstract

The terrible cost of injuries and sudden illnesses does have fatal consequences that exposes the limitations of the current prehospital processes in terms of time for emergency staff to arrive on scene and lack of first aid skills among the available incident witnesses. In this paper we aim at developing a smart pervasive chatbot for emergency case assistance based on cloud computing called SPeCECA that assists victims or incident witnesses to help avoiding deterioration of the subject's condition and maintaining his/her physical integrity until the aid arrives, which could dramatically increase the victim's survivability chances. Therefore, even a person with no first aid skills, can help the victim to survive by performing first aid support as suggested by the virtual assistant. Furthermore, thanks to its connectivity with the emergency medical service, trusted person(s), and the access to social media, SPeCECA has its own way of alarming the emergency case, in parallel, after having released the degree of the emergency situation's severity. The proposed method is a mobile pervasive healthcare service in the form of a connected mobile application as a virtual assistant for the benefit of anyone facing an emergency case. The proposed chatbot allows an online human-bot interaction that supports different scenarios for every single emergency case. The design of the system is introduced by its six interdependent components: information pre-processing component (IPPC), natural language processing component (NLPC), context component (CC), information post-processing component (IPoPC), response generator component (RGC), and alert message constructor component (AMCC).

Keywords Chatbot · Emergency · First aid · Machine learning · Pervasive health · Smart health

1 Introduction

Amongst the top 10 causes of death worldwide in 2016,[1] more than 56% were due to injuries and sudden illnesses that could have been prevented if there was an immediate medical intervention.

Indeed, when an emergency case strikes, there is no time to start thinking how to react as immediate medical intervention is required. Usually, The obvious reaction in such a situation is to call the Emergency Medical Service (EMS). However, and especially during the rush hours or in bad weather conditions, the emergency medical equipment may waste a lot of time on the road and searching for the incident's location which could dramatically decrease the survivability chances of the victim [1].

Usually, ordinary citizens, who are the first to arrive on the scene, may play vital roles in helping victims [2]. But what do people do facing an emergency case? Are these behaviours appropriate? Naturally to be ready for such situations, everyone should have first aid skills. Unfortunately, the majority of people all over the world do not get

✉ Nourchène Ouerhani
nourchne_ouerhani@yahoo.fr

Ahmed Maalel
ahmed.maalel@ensi.rnu.tn

Henda Ben Ghézela
henda.benghezala@ensi.rnu.tn

1 Higher Institute of Applied Sciences and Technology, University of Sousse, 4003 Sousse, Tunisia

2 RIADI Laboratory, National School of Computer Sciences, University of Manouba, 2010 Manouba, Tunisia

[1] Global Health Estimates 2016: Deaths by Cause, Age, Sex, by Country and by Region, 2000-2016. Geneva, World Health Organization; 2018

any first aid training, therefore, are unable to help when facing an emergency case. As for the rest, most of them do not interfere because they are prone to panic and possibly worsening the victim's situation [3].

Given the previously outlined circumstances, a smart pervasive chatbot based on cloud computing is proposed in this paper in order to assist the emergency situation and help the victim or his companions to increase his survivability chances. The proposed method is a chatbot based pervasive [4] healthcare service that offers healthcare to anyone, anytime, and anywhere. The smart [5] aspect of the chatbot plays an important role on how treating patients and how to manage disease.

The remainder of the paper is organized as follows: first, we go through the related works regarding this area, in 2. Next, we explain our approach in 3. Then, we evaluate our work by a test case in 4. And finally in 5, we conclude the work and give some future works for interesting research directions.

2 Related works

Chatbots have a long history, since 1966 [6]. But nowadays, they are experiencing a remarkable rise, thanks to the advancement of smartphones, instant messaging, artificial intelligence, and natural language processing (NLP). They became an integral part of many IT applications such as IBM's Watson,[2] Apple's Siri,[3] Google's innovation "Google Now",[4] and Microsoft's Cortana.[5] For these reasons, many researches choose to adopte chatbots into the healthcare field to avail of their multiple advantages. One of the oldest examples includes Eliza [6], a bot that simulates a psychotherapist's conversation.

More recently, Mandy [7] is a chatbot that interacts with patients using natural language in order to diagnose patient symptoms and finally generate reports. In [8], a system is designed to improve the online health model by using a chatbot to mimic human interaction in a medical situation. Its goal is to help patients choose the most appropriate path to prevent a certain disease. Regarding [9], a smart m-Health space was developed where a variety of smart services scenarios can be constructed by interaction of multiple software agents running on IoT devices. Chatbot proves their effectiveness in emergencies, for example, in [10], a mobile health service in the form of a chatbot is presented to provide user's daily health data in real time, in order to diagnose chronic diseases and to provide preventive informations and fast treatment in response to accidents that may occur in everyday life. Other chatbots were developed for different emergency cases such as Ask Diana [11], a chatbot system developed for water-related disaster management.

Concerning the emergency cases, solutions that are not based on chatbots or NLP, have been proposed, including [12], a prototype of a mobile application was proposed. It enables making emergency calls without audio communication, by selecting icons in a touchscreen mobile device. This application is important for deaf and elder people, as well as in situations of panic or some other sudden incidents that makes it difficult to articulate speech. As for [9], implements reference scenarios of personalized digital assistance for mobile patients with in emergency cases. On the other hand, a pervasive system proposed in [13] to let the emergency patient or companion, requesting for paramedic support and medical supplies via a mobile application. In [14], a system developed to provide reliable and accurate assistance to users during emergency medical situations using a drone.

Table 1 shows a comparative analysis according to several criteria: Whether it is Chatbot-based or not, The scope of the service, is it smart?, is it pervasive?, is it real time?, and is it based on cloud computing?

From the research we have done, we can conclude that despite the incessant existence of chatbots in the medical domain, few works use them to manage or assist emergency cases. That's why among them, the work of [10] is very similar to our approach except that we seek a pervasive emergency-case assistant that tries to parallelize the different actions taken in an emergency case (protecting, alerting, and avoiding deterioration of the patient's condition until the aid arrives), using mostly the pervasiveness and smartness of our system.

3 Proposed approach

The proposed SPeCECA (short of smart pervasive chatbot for emergency case assistance based on cloud computing) has two phases:

Phase 1 This phase takes place before the emergency outbreak. The first step in this phase the creation of account to get access to the chatbot. A one time login is guaranteed to facilitate the use of SPeCECA especially that we are talking about critical cases. After that SPeCECA retrieves personal user information, including general information(name, phone number, age, gender...), trusted people contacts whom will receive the alert message, social media account permissions, chatbot personaliation (prefer voice or speech output), and medical user information such as blood type and chronic disease.

[2] https://www.ibm.com/cloud/watson-assistant/

[3] https://www.imore.com/siri

[4] https://www.google.com/intl/en-GB/landing/now/

[5] https://www.microsoft.com/en-us/cortana

Table 1 Synthesis of related works

References	Chatbot-based	Scope of service	Smart	Pervasive	Real time	Cloud-based
Tsai et al. [11]	✓	Water related disaster management	✓	✗	✓	✓
Chung et al. [10]	✓	Health care service including emergency cases	✓	✗	✓	✓
Srivastava et al. [14]	Includes a chatbot in its architecture	Paramedical Emergency	✗	✗	✗	✓
Manachai [13]	✗	Prehospital Emergency	✗	✓	✓	✗
Amato et al. [8]	✓	Health On-Line Medical Suggestions	✗	✗	✗	✗
Lin et al. [7]	✓	Health care assistance	✓	✗	✓	✓
Dmitry et al. [9]	✗	Mobile health Emergency detection	✓	✗	✗	✓
Paredes et al. [12]	✗	Emergency calls without audio communication	✗	✗	✓	✗

Phase 2 During the emergency ; The user will have to activate SPeCECA by chatting with it about an emergency case. Then the bot will detect and understand there is an emergency case and must manage a whole discussion in a very minimal time in order to sustain the victim's health. Most of the important features of SPeCECA take place during this phase. SPeCECA is designed to be able to help victims or nearby perform first aid work properly: Protecting the victim and witness, sending an alert message, avoiding deterioration of the patient's condition and, maintaining his physical integrity until the aid arrives. During this phase location must be enabled by default. Amongst the supported incident type: Brutal airway obstruction, external bleeding, heart disease, open wound (cuts, scratches, puncture wounds, lacerations, penetration wounds..), trauma, and fractures.

SPeCECA's architecture is depicted in Fig. 1, we devide it into several components as follows: Information Pre-Processing Component (IPPC), Natural Language Processing Component (NLPC), Context Component (CC), Information Post-Processing Component (IPoPC), Response Generator Component (RGC), and Alert Message Constructor Component (AMCC).

More over, SPeCECA is connected to social media, trusted person(s), and the Emergency Medical Service (EMS), in order to send real time alert of emergency cases as shown in Fig. 1.

3.1 Information pre-processing component: IPPC

When the user sends a message (text, voice or image), SPeCECA pre-processes the information to extract text or tags that can be processed by other components as shown

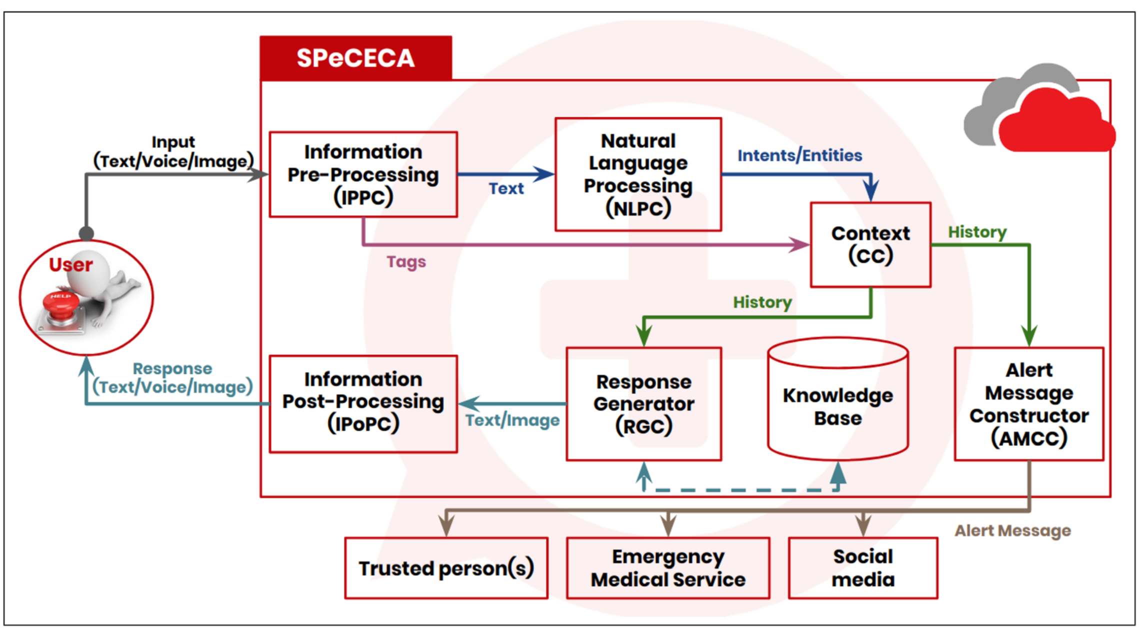


Fig. 1 SPeCECA's components and actors

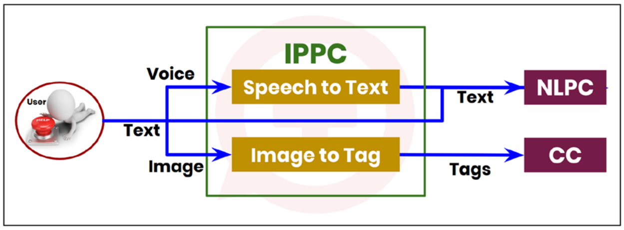


Fig. 2 Information pre-processing component (IPPC)

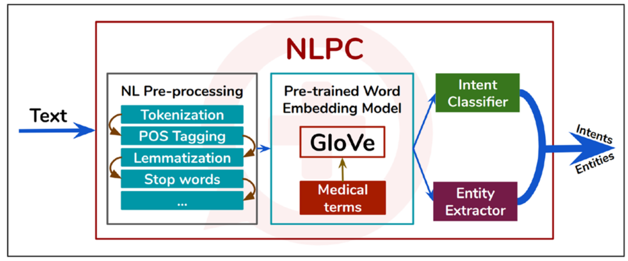


Fig. 3 Natural language processing Component (NLPC)

in Fig. 2. If the user sends a text message, IPPC brings out the text as it is to the NLPC. As for a voice input, IPPC transforms the speech to text via Speech Recognition [15]. Same as text input, the output of IPPC will be sent to NLPC. However, in case of an image as input, IPPC analyzes the image and generates tags [16] of symptoms presented such as bleeding or wound. In this case, the tags will be stored directly to the CC as no natural language processing is required for this data.

3.2 Natural language processing component: NLPC

The task of NLPC is to transform unstructured text from the IPPC and produce a structured representation composed of entities and intents, through several successive steps known as natural language pre-processing as shown in Fig. 3.

- Tokenization: is taking a text or set of text and splitting it into pieces, called tokens (words, punctuation marks, numbers, etc). Tokenization is governed by the type of the natural language [17]. In English language, words are spllited whenever there is white space, punctuation marks are treated as separate tokens.
- Part of Speech tagging (PoS tagging): is a process of marking the words in a text as corresponding to a particular part of speech, based on its definition, as well as its context. In other words, PoS tagging is assigning parts of speech (such as noun, verb, adjective, etc) to each token [18].
- Lemmatization: is "the problem of transforming a word form into its canonical form, or lemma" [19]. In fact the same word can have multiple different 'lemma's. So, based on the context it is used, the PoS tagging for the word, in that specific context, should be identified to extract the appropriate lemma.
- Stop Word Filtering: Stop words are frequently used common words. English has a lot of stop words like "and", "the", "this", and "a".

Before extracting knowledge from text, it is required to perform a transformation from the text domain to the vector domain which is referred as word embedding. In 2013, word embedding became popular when Google released a pre-trained word embeddings model called word2vec [20]. Later on, different pre-trained word embedding models appeared including GloVe [21] by Standford NLP team in 2014, fastText [22, 23] by Facebook AI Research in 2016 and BERT [24] by Google in 2018. The two most popular word embedding algorithms among them are word2vec and GloVe.

These pre-trained models have been trained on massive text corpus created from Google news and similar sources so the representations may not always transfer well to specific domains. Since GloVe counts how frequently a word appears in a context, we use GloVe as we can easily allow the chatbot to deduct which emergency case we are dealing with from the conversation with the user. Our solution is to simply train the GloVe model on our domain specific data.

Once word transformation task is done, NLPC will extract entities and classify intents that are relative to the emergency case, which will then be stored in the context component and reusable by the response generator to gather enough information to give accurate emergency instructions to perform.

In general, an entity is defined as a part of text that is of interest to the bot in order to understand the victim's state. Examples of frequently extracted entities are the type of emergency case, number of victims, injuries, victim's age, victim's gender, etc. These are only simple examples but we define custom entities for each emergency case which will be discussed further in Sect. 3.3.

Entity extraction and intent recognition requires different training techniques. Entity extraction is done by using conditional random field (CRF) [25] trained on various labeled emergency dialogues, each sentence comes with the entities and intents present. On the other hand, intent classification is done using Support Vector Machines (SVM) classifier [26] that requires only little training to make confident intent predictions, where every possible intent is encoded as a single class and predicted based on the words' feature vector. Figure 3 depicts the architecture of NLPC.

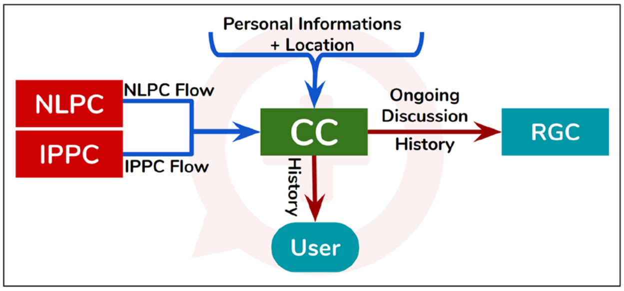


Fig. 4 Context component (CC)

3.3 Context component: CC

The Context Component is responsible for storing all information the user provided while signing up for the platform, as well as information gathered during the ongoing discussion. Therefore, we can influence how the dialogue progresses. The first thing to be stored in the CC is the user's location. Once the user starts communicating with the platform, the context component will store all intents and entities extracted, as well tags extracted from images. All these data will be consulted later on by the RGC in order to respond to the user and by the AMCC in order to construct and send the alert message, that's why the first and most important thing to store in the CC is the user's location. In other words, the CC acts as a dictionary that stores a set of key-value pairs of entities or tags extracted. This dictionary differs from an emergency case to another, thus they need to be predefined in advance for every possible case and any possible entity or tag that could be of importance to help the survivability chances of the victim (Fig. 4).

3.4 Response generator component: RGC

Before generating a response, RGC consults its knowledge base that contains predefined scenarios. These scenarios are customised for each specific emergency case that contains user's message and SPeCECA's action. It is a very challenging task because there are no public available conversational data sources. So, we had to generate training data from scratch. We had to imagine what are the most common conversations between the user and SPeCECA during a specific emergency case.

We formulate response generation task as a classification problem, over a predefined list of actions through the predefined scenarios as shown in Fig. 5. So, we choose the decision tree algorithm [27] for classification, trained on various emergency cases. Decision trees classifier was used because it imitates the human thinking so that it's so simple to understand the data and make better interpretations.

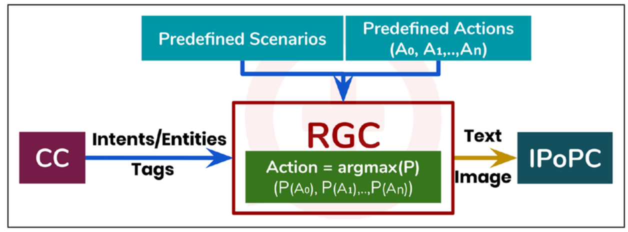


Fig. 5 Response generator component (RGC)

The answers are generated based on the information stored in CC. If there are sufficient information (i.e entities) to generate a concrete response and guide the user throughout the emergency case, the chatbot will immediately respond, otherwise, if there are some missing entities which are vital for the bot better understand the situation, it will ask the user to supply that information. This process will be repeated until the bot has enough knowledge of the emergency victim's state to generate accurate tips. The RGC outputs usually text, but also images when necessary, especially when explaining some important first aid acts that must be done accurately.

3.5 Information post-processing component: IPoPC

IPoPC is an optional component that gets triggered if the user decides to communicate with the bot in speech. The component will then transform the output text generated from RGC to speech and pass it forward to the user. If the user chose to communicate in text, IPoPC will not change the RGC's output and pass the message directly to the user as shown in Fig. 6.

3.6 Alert message constructor component: AMCC

Depending on the severity of the emergency case, SPeCECA decides whether it is mandatory to send the alert to the EMS, distribute it in the social media and even send phone message to the trusted person(s) already chosen previously by the user (Phase 1). In other words, the alert message is necessary when the situation presents high risk

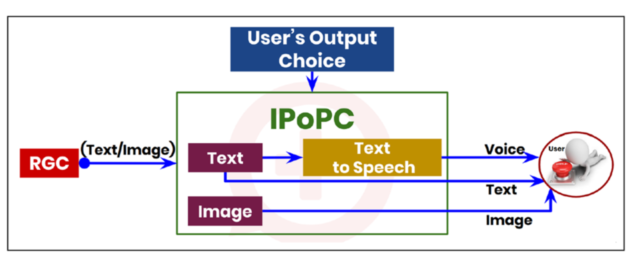


Fig. 6 Information post-processing component (IPoPC)

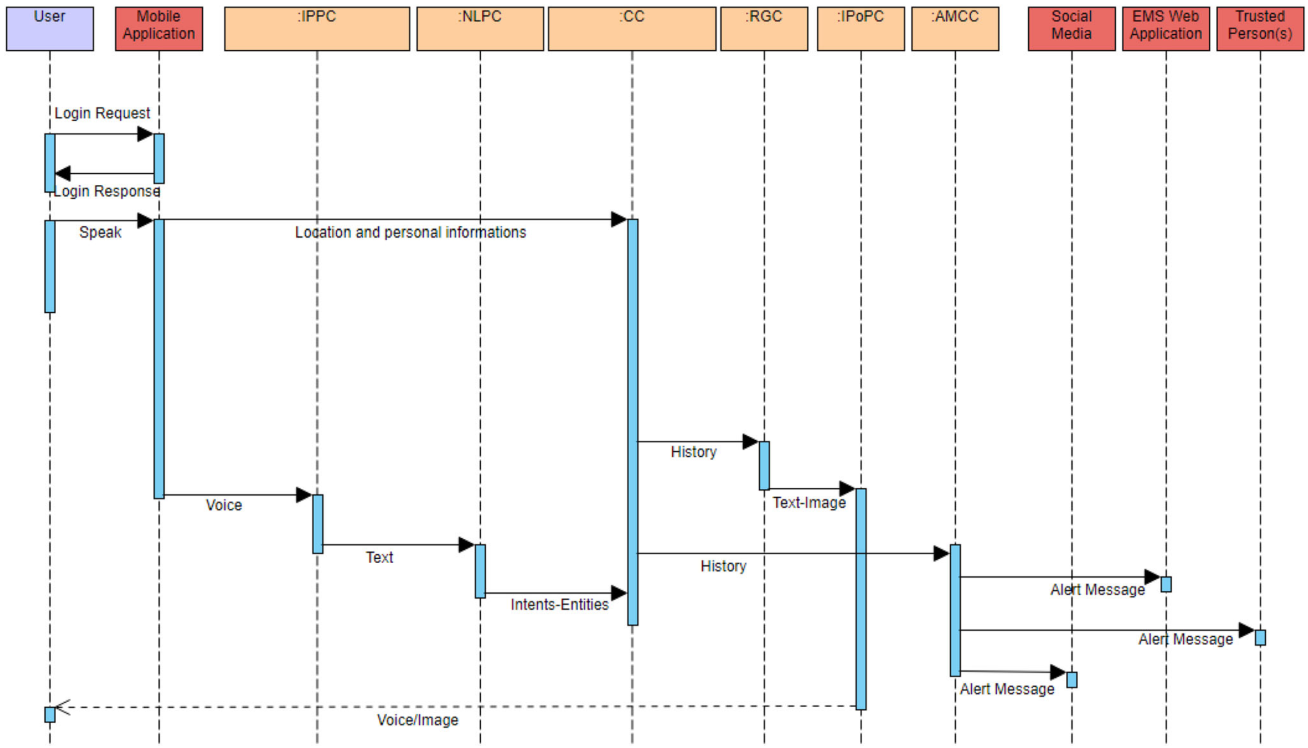


Fig. 7 Sequence diagram of a voice input flow

to general people such as road accidents. It must be fast and precise in order to minimize the delay in implementing the rescue and care chain.

Accordingly, the user is able to send a variety of form of inputs (Text/Voice/Image). Figure 7 shows a sequence diagram of a voice input flow and the contribution of every component in our architecture until the user finally receives the response from the chatbot.

4 Implementation and deployment

More than one million people die each year as a cause of road traffic accidents. Meanwhile, more than 50% of all road traffic deaths are among vulnerable road users such as pedestrians, cyclists, and motorcyclists.[6] So to evaluate our chatbot, we choose to train it to manage and assist a road traffic accident as this kind of situation requires a lot of first aid skills applied in a well-defined order.

The development of this project was done on a Computer with Intel(R) Core(TM) i5-8250U Central Processing Unit(CPU) 3.40 GHz and 8.00-GB RAM. We have tested our chatbot android application on a Smart Phone with Kirin 620 CPU, 2.00-GB RAM and 720x1280 Resolution.

The main software tools used are 64-bit Kubuntu 18.04,[7] Android Studio 6.0[8] and Visual Studio Code Version 1.32.3.[9]

The language chosen for our chatbot back-end was Python[10] version 3.6.7 for its compatibility with micro frameworks such as Flask[11] and modern deep learning libraries such as Keras.[12] We also used the open source conversational artificial intelligence framework Rasa.[13]

We develop three HTTP services following a Service Oriented Architecture (SOA) that runs on the cloud as shown in Fig. 8. The first service is *AIService* and it is dedicated to the chatbot back-end and was developed in Python and Flask web server. The second service named *BDService* is written in modern Javascript (ECMAScript 2017[14]) using NodeJS[15] because of its non-blocking asynchronous concurrent model as middle-ware between client and our database. The last service is called

[6] https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries

[7] https://kubuntu.org/

[8] https://developer.android.com/studio/

[9] https://code.visualstudio.com/

[10] https://www.python.org/downloads/release/python-367/

[11] http://flask.pocoo.org/

[12] https://keras.io/

[13] https://rasa.com/docs/

[14] http://www.ecma-international.org/ecma-262/8.0/index.html

[15] https://nodejs.org/en/

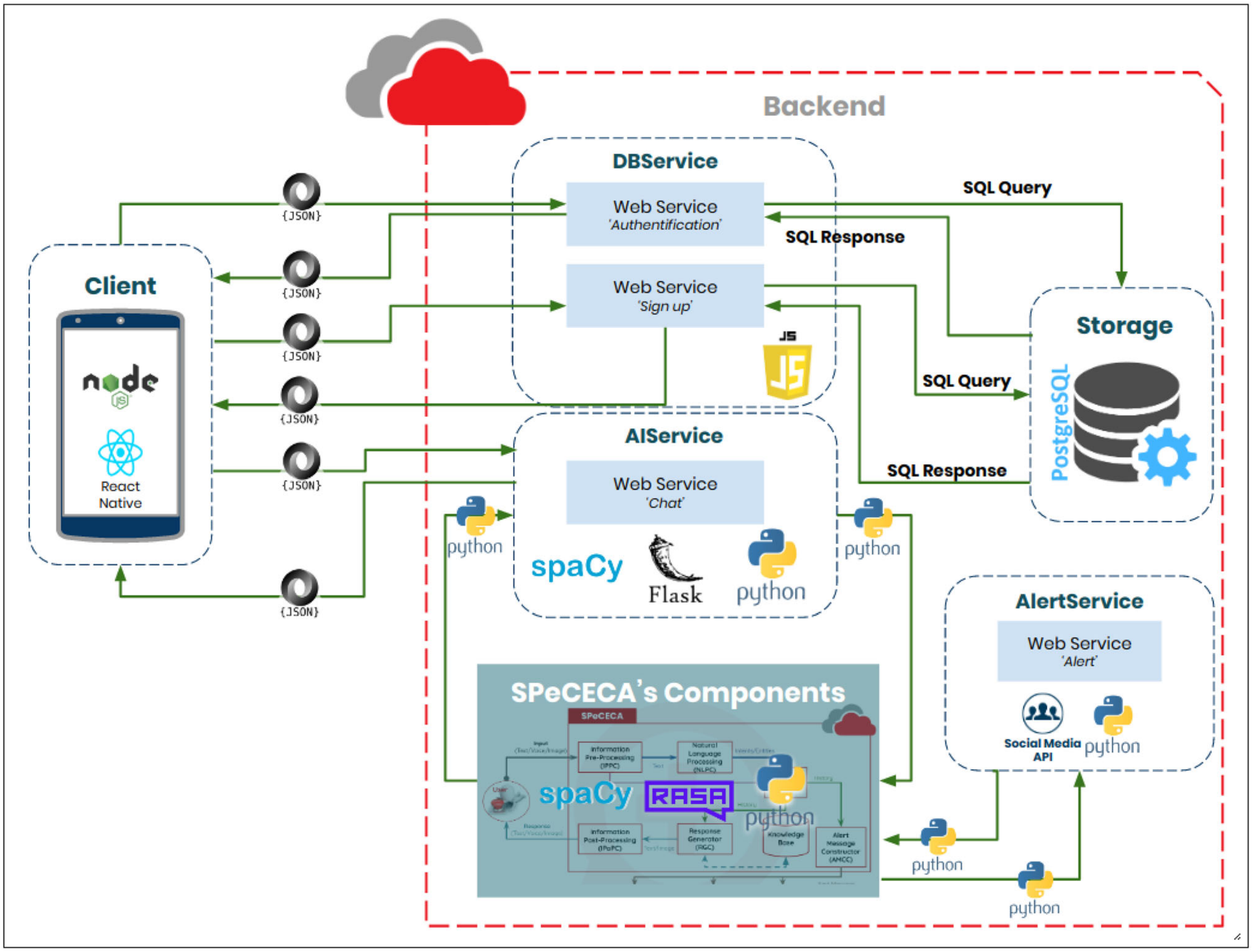


Fig. 8 SPeCECA's data flow architecture

AlertService and is designed in Python to take advantage of the power of this language in the integration of different social media API. This service guarantees the spread of alert message on social media networks.

Since our solution is cloud based, we have developed 2 separate but inter-dependent applications: The mobile application that will be used by the user and the cloud based platform, where each of proposed components is written as web services as shown in Fig. 8.

Our web services are exposed as REST API micro services on the cloud, referenced via their URI.

As mentioned in Sect. 3.2, the NLPC deals with training a chatbot to understand user inputs. So to be able to achieve this task, we used the SVM classifier which loads pre-trained language model which then is used to represent each word in the user message as word embedding.

Our next task is to teach SPeCECA to respond to messages by training the RGC. The training data for RGC is called scenarios. The user's inputs are expressed as intents as well as the corresponding entities, and the chatbot responses are expressed as actions. Finally, we will train RGC citing the policies that should be used to train it.

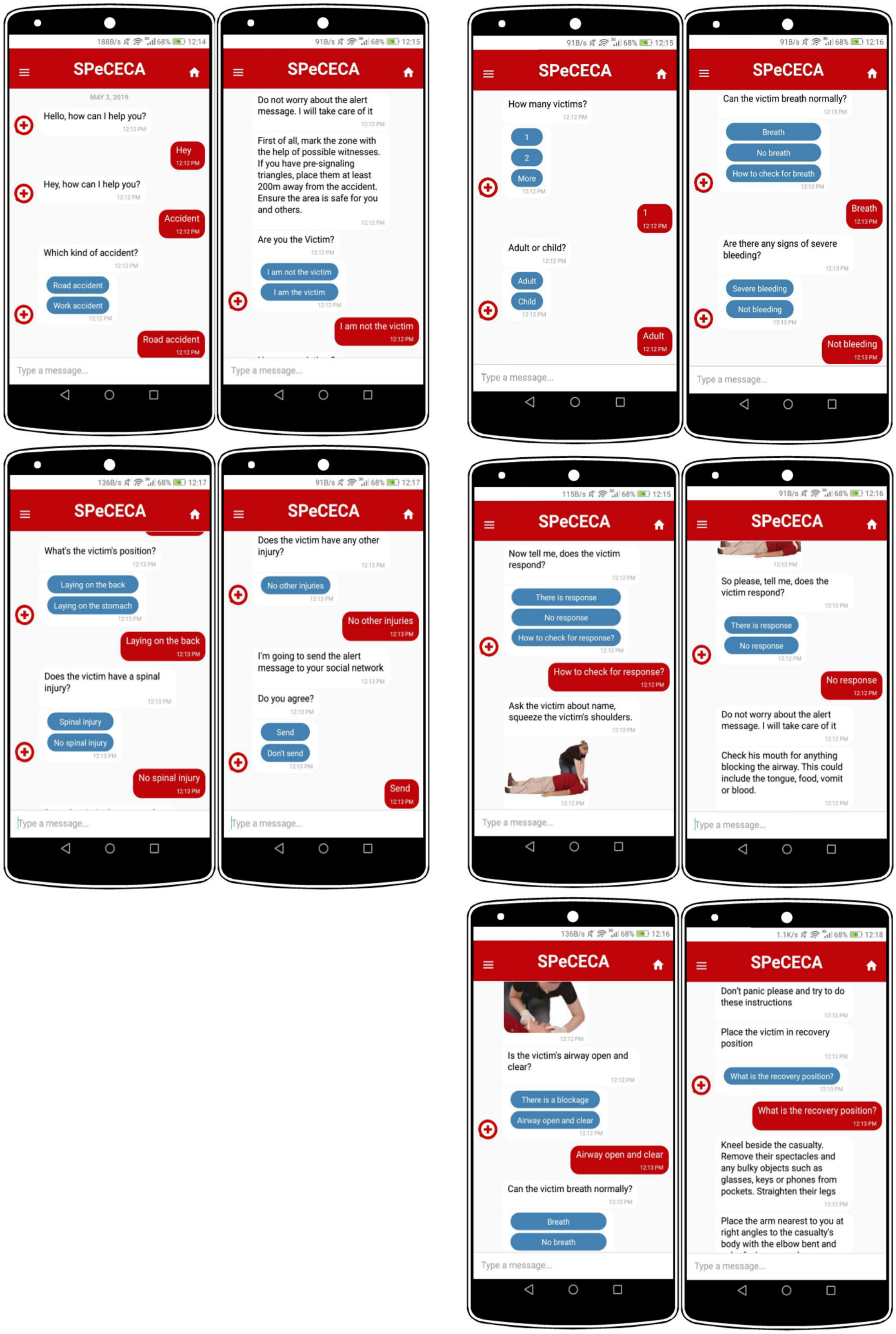


Fig. 9 SPeCECA's responses amongst a road traffic injury case: road accident involving pedestrians. In this case the user has chosen to communicate with SPeCECA with text. The bot keeps requesting information from the user so it can give helpful first aid tips

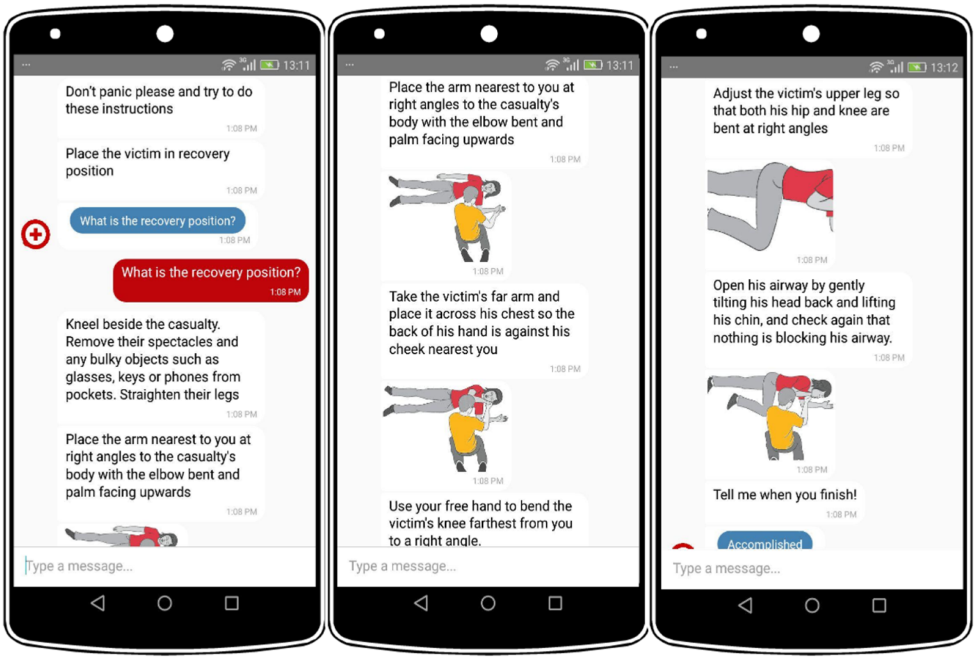


Fig. 10 SPeCECA explains to the user the recovery position step by step

To sum up, SPeCECA asks specific questions to understand the victim's condition. Once the determination is complete, first aid suggested by SPeCECA is performed before aid arrives. An example of implemented process is shown in Fig. 9.

The user is free whether to write his own answer or to choose one of the answers suggested by the chatbot, this method allows a faster response time from the user.

Some of the responses from the bot may include some emergency vocabulary which the user may be unfamiliar with, thus the chatbot explains some medical acts through images such as "The recovery position" as demonstrated in Fig. 10 In parallel and at a definite moment, SPeCECA delivers the information and conditions of the victim to the EMS, in social media, and to the trusted person(s).

As in Fig. 11 the user is able to sign up to our platform, enter his credentials and private information that will be shared with our cloud-based chatbot during any future conversation. Upon an emergency case, the user can request help from the bot by providing relevant information to the bot, either through speech, text, or selecting choices suggested by the chatbot, which upon collecting enough data will suggest the required treatment process to the user.

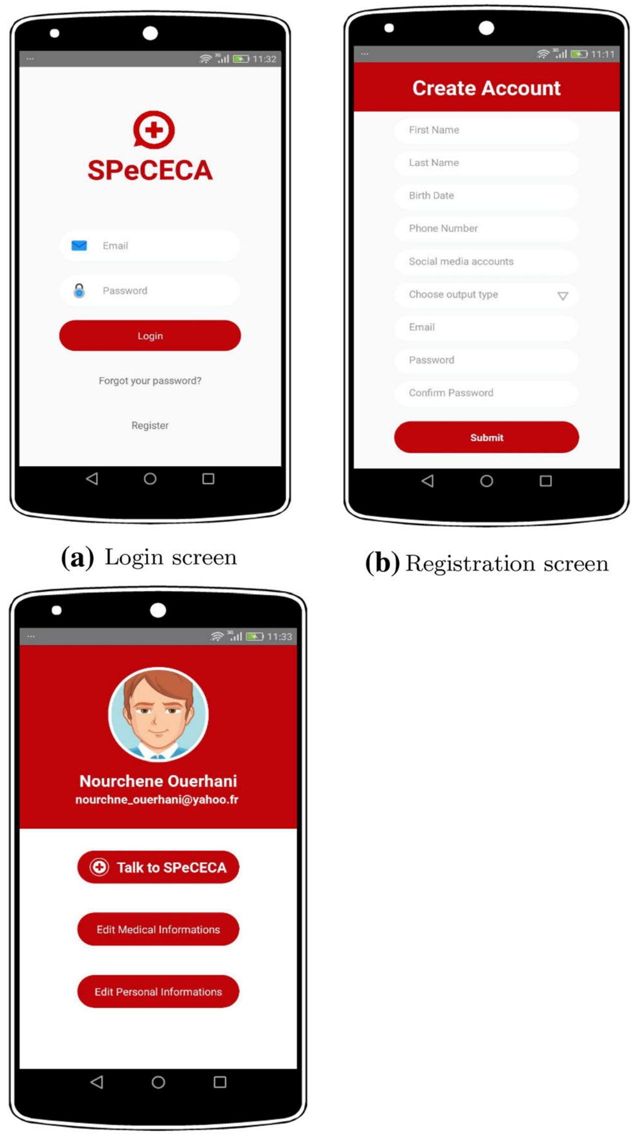
As for the alert message, Fig. 12 shows an example sent in Social media related to our case study.

5 Conclusions and future works

We develop a smart pervasive chatbot for emergency case assistance based on cloud computing, called

SPeCECA who is designed to help people react well in emergency cases. We use GloVe to solve the NLP problem in our case because it works well according to our evaluation experiments. Much further work is needed to improve SPeCECA's capability to include more diseases.

Fig. 11 SPeCECA's screens



(a) Login screen

(b) Registration screen

(c) Home screen

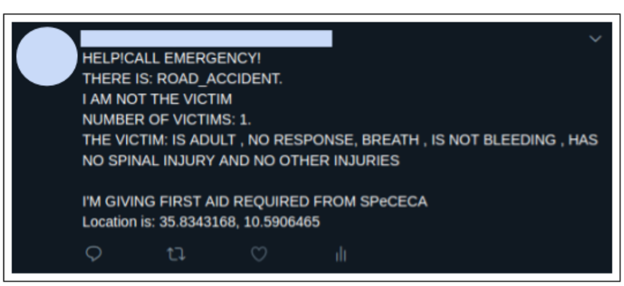


Fig. 12 The alert message sent on a social media platform

Additionally, in order to determine whether it is a true emergency alert or not, we recommend the emotion detection by facial expressions or voice analysis [28–30] and give EMS more analytical insights and predictions of possible future emergency cases [31].

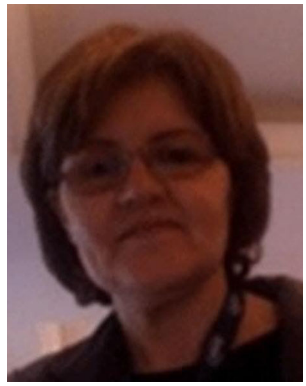
References

1. Söderström, E., van Laere, J., Backlund, P., Söderholm, H.M. , editor="Johansson, Björn and Andersson, Bo and Holmberg, Nicklas: Combining Work Process Models to Identify Training Needs in the Prehospital Care Process, Perspectives in Business Informatics Research, 375–389, Springer International Publishing (2014)
2. Whittaker, Joshua, McLennan, Blythe, Handmer, John: A review of informal volunteerism in emergencies and disasters: definition, opportunities and challenges. Int. J. Disaster Risk Reduct. **13**, 358–368 (2015). https://doi.org/10.1016/j.ijdrr.2015.07.010

3. Pellegrino, J., Oliver, E., Orkin, A., Marentette, D., Snobelen, P., Muise, J., Mulligan, J., De Buck, E.: A call for revolution in first aid education. Int. J. First Aid Educ. **1**, 5 (2017). https://doi.org/10.21038/ijfa.2017.0001
4. Varshney, Upkar: Pervasive healthcare and wireless health monitoring. Mobile Netw. Appl. **12**, 113–127 (2007). https://doi.org/10.1007/s11036-007-0017-1
5. Penmatsa, P.L., Rama Kkoti Reddy, D.V.: Smart Detection and Transmission of Abnormalities in ECG via Bluetooth, 2016 IEEE International Conference on Smart Cloud (SmartCloud), 41-44 (2016)
6. Weizenbaum, J.: ELIZA&Mdash;a Computer Program for the Study of Natural Language Communication Between Man and Machine, Commun. ACM, Vol. 9, pp. 36–45, ACM, New York (1966). https://doi.org/10.1145/365153.365168
7. Ni, L., Lu, C., Liu, N., Liu, J.: MANDY: towards a smart primary care Chatbot application. In: International Symposium on Knowledge and Systems Sciences, pp. 38–52 (2017). Springer, Singapore, https://doi.org/10.1007/978-981-10-6989-5_4
8. Amato, F., Marrone, S., Moscato, V., Piantadosi, G., Picariello, A., Sansone, C.: Chatbots Meet eHealth: Automatizing Healthcare (2017)
9. Korzun, D.G., Borodin, A.V., Timofeev, I.A., Paramonov, I.V., Balandin, S.I.: Digital assistance services for emergency situations in personalized mobile healthcare: Smart space based approach, In: Proceedings of the 2015 International Conference on Biomedical Engineering and Computational Technologies (SIBIRCON), 62–67 (2015). https://doi.org/10.1109/SIBIRCON.2015.7361852
10. Chung, K., Park, R.: Chatbot-based heathcare service with a knowledge base for cloud computing. Clust. Comput. (2018). https://doi.org/10.1007/s10586-018-2334-5
11. Tsai, M.-H., Chen, J.Y., Kang, S.-C.: Ask Diana: A Keyword-Based Chatbot System for Water-Related Disaster Management. Water **11**, 234 (2019). https://doi.org/10.3390/w11020234
12. Paredes, H., Fonseca, B., Cabo, M., Pereira, T., Fernandes, F.: SOSPhone: a mobile application for emergency calls. Univ. Access Inf. Soc. **13**, 277–290 (2014). https://doi.org/10.1007/s10209-013-0318-z
13. Toahchoodee, M.: ARSA-the pervasive Rescuer Supporting System for the Pre-hospital Emergency Medical Service, In: Proceedings of the 2017 14th International Joint Conference on Computer Science and Software Engineering (JCSSE), 1–6 (2007). https://doi.org/10.1109/JCSSE.2017.8025920
14. Srivastava, M., Suvarna, S., Srivastava, A., Bharathiraja, S.: Automated emergency paramedical response system. Health Inf. Sci. Syst. **6**, 22 (2018). https://doi.org/10.1007/s13755-018-0061-1
15. Mohammed, Mohssen, Khan, M.B., Bashier, E.B.M.: Machine Learning: Algorithms and Applications. CRC Press, Boca Raton (2016)
16. Zhang, Y., Gong, B., Shah, M.: Fast Zero-Shot Image Tagging, CoRR, abs/1605.09759 (2016). arxiv: abs/1605.09759
17. Webster, J.J., Kit, C.: TOKENIZATION AS THE INITIAL PHASE IN NLP, COLING 1992 Volume 4: The 15th International Conference on Computational Linguistics (1992) http://www.aclweb.org/anthology/C92-4173
18. Kumar, D., Josan, G.S.: Part of speech taggers for morphologically rich Indian languages: a survey. J. Comput. Appl. **6**(5), 32–41 (2010)
19. Eger, S., Gleim, R., Mehler, A.: Lemmatization and Morphological Tagging in German and Latin: A Comparison and a Survey of the State-of-the-art, LREC (2016)
20. Mikolov, T., Chen, K., Corrado,G., Dean, J.: Efficient Estimation of Word Representations in Vector Space, CoRR, abs/1301.3781, (2013). arxiv: abs/1301.3781
21. Pennington, J., Socher, R., Manning, C.: Glove: Global Vectors for Word Representation, In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 1532–1543, Association for Computational Linguistics, Doha, Qatar (2014) https://doi.org/10.3115/v1/D14-1162, http://aclweb.org/anthology/D14-1162
22. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching Word Vectors with Subword Information, CoRR, abs/1607.04606 (2016) arXiv:abs/1607.04606
23. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of Tricks for Efficient Text Classification, CoRR, abs/1607.01759 (2016) arxiv: abs/1607.01759
24. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, CoRR, abs/1810.04805 (2018). arxiv: abs/1810.04805
25. Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data, In: Proceedings of the Eighteenth International Conference on Machine Learning, 282–289, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2001). http://dl.acm.org/citation.cfm?id=645530.655813
26. Chang, C.-C., Lin, Chih-Jen: LIBSVM: A library for support vector machines. ACM TIST **2**, 27:1–27:27 (2011)
27. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco (1993)
28. Zhang, Y., Chen, M., Huang, D., Di, W., Li, Y.: iDoctor: Personalized and professionalized medical recommendations based on hybrid matrix factorization. Futur. Gener. Comput. Syst. **66**, 30–35 (2017). https://doi.org/10.1016/j.future.2015.12.001
29. Tivatansakul, S., Ohkura, M., Puangpontip, S., Achalakul, T.: Emotional healthcare system: Emotion detection by facial expressions using Japanese database, 2014 6th Computer Science and Electronic Engineering Conference, CEEC 2014—Conference Proceedings, pp. 41–46 (2014). https://doi.org/10.1109/CEEC.2014.6958552
30. Hossain, M.S., Muhammad, G.: An emotion recognition system for mobile applications. IEEE Access **5**, 2281–2287 (2017). https://doi.org/10.1109/ACCESS.2017.2672829
31. Bennani, S., Maalel, A., Ghézala, H.B., Abed, M.: Towards a decision support model for the resolution of episodic problems based on ontology and case bases reasoning: application to terrorism attacks. In: IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA), Hammamet, pp. 1502–1509 (2017)

Nourchène Ouerhani is a PhD student in computer science at RIADI laboratory, National School of Computer Sciences (ENSI – Tunisia). She has a Master degree in Computer Science in Smart Pervasive Systems from Higher Institute of Applied Science and Technology of Sousse (ISSATSo), University of Sousse, Tunisia.

Ahmed Maalel has a PhD in Computer Science. He is currently assistant professor at Higher Institute of Applied Science and Technology of Sousse (ISSATSo), University of Sousse, Tunisia. He is a Researcher at RIADI laboratory, National School of Computer Sciences (ENSI – Tunisia). His research interests are related to these topics: Machine Learning, Support Decision Systems, Knowledge Management and Semantic Web. He served as technical chair, TPC member and reviewer for many leading international CS conferences and journals. He is IEEE Senior Member and Co-Founder and President of Association of Scientific Research and Innovation in Computer Science – ARSII.



Henda Ben Ghézela is currently Professor of Computer Science at National School of Computer Sciences of Tunis. Her research interests lie in the areas of information modeling, databases, temporal data modeling, object-oriented analysis and design, requirements engineering and specially change engineering, method engineering. She is Director of the RIADI labs., ENSI, University of Manouba.