

This is a self-archived version of an original article. This version may differ from the original in pagination and typographic details.

Author(s): Khriyenko, Oleksiy; Rönkkö, Konsta; Tsybulko, Vitalii; Piik, Kalle; Le, Duc Pham Minh; Riipinen, Tommi

Title: Stroke Cognitive Medical Assistant (StrokeCMA)

Year: 2018

Version: Published version

Copyright: © The Author(s) 2018. This article is published with open access by the GSTF

Rights: CC BY-NC 3.0

Rights url: <https://creativecommons.org/licenses/by-nc/3.0/>

Please cite the original version:

Khriyenko, O., Rönkkö, K., Tsybulko, V., Piik, K., Le, D. P. M., & Riipinen, T. (2018). Stroke Cognitive Medical Assistant (StrokeCMA). *GSTF Journal on Computing*, 6(1).
https://doi.org/10.5176/2251-3043_6.1.112

Stroke Cognitive Medical Assistant (StrokeCMA)

Oleksiy Khriyenko
*Faculty of Information
 Technology*
University of Jyväskylä
 Jyväskylä, Finland
 oleksiy.khriyenko@jyu.fi

Konsta Rönkkö
*Faculty of Information
 Technology*
University of Jyväskylä
 Jyväskylä, Finland
 konsta.t.ronkko@student.jyu.fi

Vitalii Tsybulko
*Faculty of Information
 Technology*
University of Jyväskylä
 Jyväskylä, Finland
 tsybulkovitaliy@gmail.com

Kalle Piik
*Faculty of Information
 Technology*
University of Jyväskylä
 Jyväskylä, Finland
 kaanpiik@student.jyu.fi

Duc Pham Minh Le
*Faculty of Information
 Technology*
University of Jyväskylä
 Jyväskylä, Finland
 miduleph@student.jyu.fi

Tommi Riipinen
*Faculty of Information
 Technology*
University of Jyväskylä
 Jyväskylä, Finland
 tommi.riipinen@gmail.com

Abstract— Stroke is the number two killer after heart disease since it is responsible for almost 10% of all deaths worldwide. The main problem with a stroke is a significant delay in treatment that happened mainly due to inappropriate detection of stroke symptoms or inability of a person to perform further necessary actions, and might cause death, permanent disabilities, as well as more expensive treatment and rehabilitation. Nowadays assessment of a stroke is done by human, following widely adopted FAST approach of stroke assessment. Since a human factor become one of the causes of treatment delay, offered solution will try to minimize this factor. Artificial Intelligence, Cognitive Computing, Machine Learning and Data Mining, NLP and other technologies make possible to elaborate a smart solution that enable automated stroke symptoms detection on earlier stages without self-assessment or assistance of another person, solution that in time provides notification to corresponding caregivers (family members, responsible medical worker, etc.) and even able to directly call emergency, explaining the cases and providing all necessary evidences to support further decision making. Thus, the paper presents feasibility study of IBM Watson cognitive computing services and tools to address the issue of automated stroke symptoms detection to elaborate smart supportive tool in the pocket of people under high risk of a stroke attack.

Keywords— *cognitive computing; medical assistant; decision support system; stroke symptoms detection; automated diagnostics; natural language processing; IBM Watson.*

I. INTRODUCTION

Nowadays, we are dealing with a huge and constantly growing volume of data. To smartly utilize all the collected data, improve productivity and increase competitiveness in various field of life, human requires decision making support systems that efficiently process and analyze the data and significantly speed up this process. A number of supervised and unsupervised Machine Learning and Data Mining techniques exist to help us to deal with structured data. But in a real life, we pretty much deal with unstructured data. It is a data, where powerful knowledge and valuable information are hidden inside human-readable plain texts, images, audio and

video materials. To address this type of data, machines have to be able to understand it not only on syntactical, but, what is even more important, on a level of semantics that data contains. Therefore, big players such as IBM, Google, Microsoft, Facebook, Intel as well as variety of SMEs are working on elaboration and offer different Cognitive Computing¹ services and tools that heavily utilize Natural Language Processing² (NLP), Semantic Web [1] and Linked Data [2], and Deep Learning³ technologies to get a value from unstructured data to be used in decision making support. With respect to application areas and domains where decision making support systems would be useful and helpful, the scope is unlimited. It could be any area, where we have a huge amount of logged data and profiles, collected feedbacks and observation measurements, data produced by IoT devices and social web content produced by human, human driven statements and guidance, mass media content or other data that could be processed to retrieve and infer hidden knowledge. Businesses and industries, health and social care, education and science, sport and wellbeing, security, transportation and other areas of human life are lacking Artificial Intelligence (AI) based solution to facilitate decision support.

This work addresses smart cognitive computing based solutions in healthcare domain and particularly, a solution for people who are under the high risk of a stroke attack. Stroke is the number two killer after heart disease since it is responsible for almost 10% of all deaths worldwide. With respect to World Health Organization⁴ (WHO) report 2014 and related studies [3], there are approximately 17 million people having a stroke each year worldwide. About one third of them die because of it and another third are living with permanent disabilities. According to forecast, one of six people around the world will have a stroke in their life. One of the main problems with this respect is a significant delay in treatment that happened mainly

¹ https://en.wikipedia.org/wiki/Cognitive_computing

² https://en.wikipedia.org/wiki/Natural-language_processing

³ https://en.wikipedia.org/wiki/Deep_learning

⁴ <http://www.who.int>

due to inappropriate detection of stroke symptoms or inability of a person to perform further necessary actions, and might cause death, permanent disabilities, as well as more expensive treatment and rehabilitation. By reducing amount of delayed cases we may save lives and help to avoid harmful consequences for health conditions of the users.

Nowadays assessment of a stroke is done by human, following widely adopted FAST approach of stroke assessment. Since a human factor become one of the causes of treatment delay, offered solution will try to minimize this factor. Applying Artificial Intelligence, Cognitive Computing, Machine Learning and Data Mining, NLP, as well as related tools provided by IT giants as IBM, Google and Microsoft it become feasible to elaborate a solution that enable automated stroke symptoms detection on earlier stages without self-assessment or assistance of another person, solution that in time provides notification to corresponding caregivers (family members, responsible medical worker, etc.) and even able to directly call emergency, explaining the cases and providing all necessary evidences to support further decision making.

Therefore, the main objective of this work is to make a feasibility study of existing cognitive computing tools and services (and IBM Watson cognitive services in particular) and check their applicability to tackle the mentioned problem. IN the next section we highlight main stroke indicators that are used in elaborated solution. Section 3 presents main approached being used to automate detection of stroke symptoms. Prototype implementation aspects and limitations are covered in the Section 4. Following section shows value and impact of the offered solution. Finally, two last sections refer to related works and conclude current work with the planes on further development.

II. STROKE INDICATORS

For each minute a stroke goes untreated and blood flow to the brain continues to be blocked, a person loses about 1.9 million neurons. This could mean that a person's speech, movement, memory, and so much more can be affected. Stroke symptoms include sudden:

- one side numbness/weakness of face, arm or leg;
- confusion, trouble speaking, or understanding;
- trouble seeing in one or both eyes;
- trouble walking, dizziness, loss of balance/coordination;
- severe headache with no known cause.

In the App, stroke recognition will be based on three the most functional indicators: slurred and/or confused speech, one-side facial and arm weakness affected by a stroke. In such a way, App follows well known FAST way to assist the person under a stroke (see Fig. 1):

- **F (FACE):** Ask the person to smile. Does one side of the face droop?);
- **A (ARM):** Ask the person to raise both arms. Does one arm drift downward?);
- **S (SPEECH):** Ask the person to repeat a simple phrase. Is their speech slurred or strange?);
- **T (TIME):** If you observe any of these signs, call the ambulance immediately).

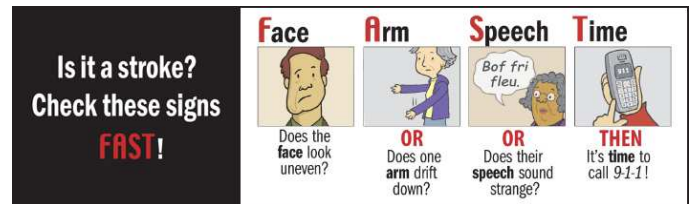


Figure 1. FAST way to assist the person under a stroke⁵.

If any 1 of these 3 signs is abnormal, the probability of a stroke is 72% [4].

Talking about speech-based symptoms, we may highlight two problems that usually occur either simultaneously or separately depending on the localization and size of damaged area of the brain. They are so called “nonsense” and “slurring” problems. Having the first one, affected by stroke person may use nonsense word or phrase during the conversation, or even completely change the topic and talk about irrelevant to the conversation context things. The second problem (slurring of speech) is caused by affected muscles that help human to correctly produce a sound. Thus, person becomes incapable to correctly pronounce the words and starts to slur them. To detect this, some special control phrase(s) (e.g. “you can’t teach an old dog new tricks”) are usually used in human-driven assessment. Regarding facial indicator of the stroke, it might be enough to ask person to show his/her teeth. So called “smile test” is used to check for one-sided facial weakness - a classic sign of stroke. Even if the test is named “smile test”, it is better to ask person to “grin” (show the teeth) rather than smile. In this case, affected by stroke person will have obvious asymmetry in the lips’ corners - one side usually happens to be dropped down. Since stroke might impact not only on facial muscles, but on any other muscles of the body as well. Usually it influence on one-side of the human body making muscles of that side much weaker than another. Therefore, detection of asymmetry in arms movement capabilities considered as another very effective indicator of stroke recognition. Sometimes it might be enough to check whether person is not able to raise a hand or hold it for a while in certain position (e.g. “up” or “in front of” position), but it is more reasonable to compare such performance of one hand against the other. Thus, big difference (“delta”) between physical capabilities of two arms of a person might be considered as a possible sign of a stroke.

III. STROKE COGNITIVE MEDICAL ASSISTANT

Stroke Cognitive Medical Assistant (StrokeCMA) - is going to be a smart supportive system (mobile application (further “App”) and corresponding back-end solution on a cloud) for the people under high risk of a stroke that ensures fast medical help reducing harmful delay in treatment. Knowing the signs and symptoms of a stroke is the first step to ensuring medical help is received immediately. With the help of the StrokeCMA, identification of the signs and symptoms is automated and could speed up a process of stroke recognition. As for any other supportive systems, for StrokeCMA it is important that solution does not disturb user much, making

⁵ <http://www.mass.gov/eohhs/gov/departments/dph/programs/communitiy-health/heart-disease-stroke/stroke-heroes-act-fast.html>

him/her willing to actively use it. Therefore, to be attractive, the StrokeCMA is implemented in a form of a smart and cognitive enough *Digital Artificial Friend* that unobtrusively communicates with user as a friend does. From time to time (according to the scheduling settings, as well as, depending on other contextual information) StrokeCMA calls a user and, via unobtrusive friendly natural conversation, snapped face images and some other captured sensor data of the phone, performs diagnostics recognizing early symptoms and signs. App could be also activated by user intentionally (if user fills unconfident regarding the health conditions) or automatically during the phone call made/received by user.

Since one of the main features of the App is unobtrusiveness, logic of the App is split to two modes: "Friend" mode (unobtrusive weak stroke indicators recognition) and "Doctor" mode (more comprehensive stroke detection). In the first mode App tries to recognize possible indicators via natural conversation with user without any specific questions and requests for specific actions. Here App asks user some simple questions about the mood, his/her plans for the day with respects to the various topics (e.g. family, job, hobbies, health, etc.), pretending that user has a conversation with the old friend. If some weak signals of a stroke are recognized in "Friend" mode, App switches into "Doctor" mode where asks user to perform more specific actions to make more precise analysis. Communication with user is mainly done in voice based mode (using speech-to-text and text-to-speech transformation services). However, App is additionally facilitated with visual text, image, video/animation support for more convenient guidance. Since "Doctor" mode requires some data (e.g. control samples), initial collection of such required data is done in "Setup" mode of the App - mode when user runs App for the first time and is asked to provide some initial data and perform certain actions. With respect to Cognitive Computing tools, we use Watson Cognitive Computing services⁶ available from IBM Bluemix cloud as well as other tools and open source libraries.

A. Face

To be unobtrusive and still constantly observe and analyze possible facial indication of a stroke, App constantly keeps "eye" contact with the user. Therefore, user should look at phone (App's GUI) like at a mirror and try to place the face inside special area in App's "mirror". If App loses "eye" contact with the user, it asks him/her to keep contact back. This way, App is able from time to time to take snapshots and check for possible weak signs of a stroke, as well as, store snapped images of "healthy" user as a training set ("general" category) to be used for retraining a face-based classification model to detect a stroke signals. App keeps "eye" contact not only during the speech-based stroke indicators recognition, but also during a conversation prior to start arm weakness detection.

Since snapshots are taken in unpredicted by user moment, there will be faces of a speaking person. Thus, this "general" training set will be quite far from a good set to provide high detection performance. But App uses this set only in the "Friend" mode to meet the requirement for unobtrusiveness

and still be able to detect weak signals of a stroke. As soon as certain following snapshot is classified as different from "healthy" face with confidence level higher than certain threshold, App is switched to the "Doctor" mode.

In the "Doctor" mode, App asks user to take two different types of snapshots of user's face: "grinning" (showing the teeth) and "normal" face. At the same time, some snapshots of "general" category face are taken during conversation. These pictures are classified either based on image classification models trained on corresponding sets of the same type/category snapshots taken before (samples of "healthy" face), or using landmark based asymmetry detection approach. Finally, aggregated analysis of classification confidences gives us more precise probability of a stroke impact. To train corresponding image recognition models, we have to have sufficient amount of user's images regarding those different categories. Therefore, to populate the training sets with new samples of "healthy" user (in addition to the pictures initially taken in "Setup" mode), in the "Friend" mode, from time to time, App asks the user to show those two types of faces and takes snapshots of corresponding categories as well as "general" one. Additionally, new samples collection allows classification models to take into account possible changes of a face with a time, as well as to update control measurements for landmark based classification approach. However, collection process might bring some obtrusiveness to the "Friend" mode, but so far there is no other way to collect refreshed training sets.

The main challenge is to find an optimal way to detect face-based stroke indicators. The first naïve approach is to try to classify taken snapshot against a general classification model built based on a big (not user specific) training sets of "healthy" faces and "stroke affected" faces of people. Having a big enough training samples for both classes we may let a neural network (used behind Watson Visual Recognition service) to select important features and to catch a common pattern that is sensitive to stroke signs. It might be difficult to find big enough set of faces affected by a stroke, since usually there is no time to take such a pictures when stroke is detected. Nevertheless, for this purpose we may also use faces of people who have "Bell's palsy" (facial palsy) disease⁷, since it is the most common cause of facial paralysis. Visually, such faces look pretty similar to the faces affected by a stroke. As a future research work towards improvement of this functionality, we may consider elaboration of customized architecture of a deep neural network that fits our specific needs of classification better than architecture hidden behind a general Watson Visual Recognition service. It is quite tricky to use this naïve approach and make model more personalized, since mostly we lack a training set with sufficient amount of "stroke affected" faces of a user (see Fig. 2). As an option, we may use sets of artificially created images of "stroke affected" faces for "grinning", "normal", as well as "general" face categories, to build corresponding classification models. Practically it might be possible to simulate a "stroke affected" face using some medication that are used in dental treatment and cause numbness of a face muscles as a side effect. Further,

⁶ <https://www.ibm.com/watson/developercloud/services-catalog.html>

⁷ https://en.wikipedia.org/wiki/Bell%27s_palsy

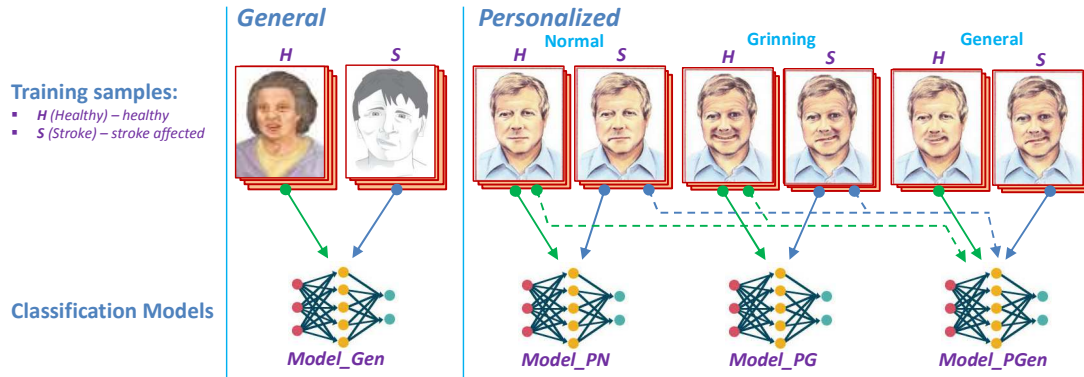


Figure 2. Naïve classification.

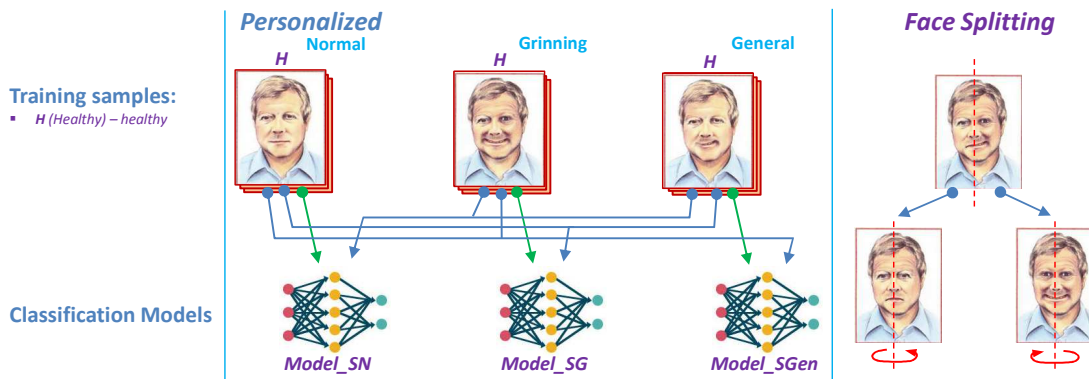


Figure 3. Symmetry based classification.

classification models could be improved as soon as we get more real examples of “stroke affected” faces. Alternative approach is to try to recognize stroke signals based on “healthy” faces of a user only. Here we build classification model based on two training sets: images that belong to corresponding face category and images that belong to other categories (see Fig. 3). Since stroke causes one-side face weakness and brings asymmetry by dropping it, to recognize a stroke signs we have to detect sufficient difference between left and right sides of the snapped image. Therefore we split this image vertically into two halves and extend each half with corresponding symmetrical “mirror” copy. Prior splitting, we have to pre-process the image to be sure that it is correctly vertically calibrated. Now we are ready to classify these two images against corresponding category it has been snapped for (“grinning”, “normal” and “general”) using Watson Visual Recognition service. The assumption is that difference between classification confidences is higher than certain threshold if the original snapshot belongs to “stroke affected” faces. Corresponding thresholds are defined empirically based on simulations. With a purpose to take into account possible initial asymmetry of a face, generalized initial difference between confidences is calculated and applied to the threshold. Having snapshots for all face categories we may aggregate the final confidence of stroke signs recognition. In comparison to the previous (naïve classification) approach, symmetry based classification does not require any training set of stroke affected faces. Thus, it could be considered as a more appropriate approach for the first prototype.

For the above mentioned image classification based approaches we may use some alternative to the Watson Visual Recognition service tools. For example, Google Cloud Vision API⁸, Microsoft Custom Vision⁹, Clarifai¹⁰, as well as OpenCV¹¹ and other image processing library could be used to perform same image classification. Since the library might be used locally on a smartphone, face based stroke indicator detection may be independent of internet access. It might be very beneficial in case user does not have internet connection or connection is slow.

Talking about face asymmetry detection, we may also use image recognition tools designed specifically for face detection/recognition/classification. Since one-side drop of lips’ corner is one of the most recognizable signs of a stroke, face processing techniques, which take into consideration so called facial landmarks - points of a human face (e.g. centers of eyes, nose, corners of lips, etc.), are able to give us a promising result. Here we may highlight two directions to proceed. We may directly use existing face similarity recognition services for face splitting based approach (mentioned above) and compare similarity confidence of two generated faces. Usually, such services utilize similarity metrics based on relations of various distances between certain landmarks of a face. They

⁸ <https://cloud.google.com/vision/>

⁹ <https://customvision.ai/>

¹⁰ <https://www.clarifai.com/>

¹¹ <http://opencv.org/>

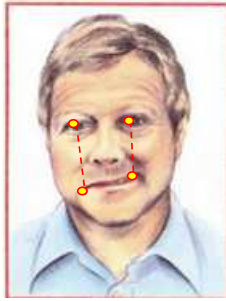


Figure 4. Landmark based asymmetry detection.

might work well to recognize the faces that belong to the same person, but it does not mean that they are perfectly able to catch specific face asymmetry caused by a stroke. Another approach is to directly use positions of the most relevant landmarks and calculate the differences with respect to different sides of the face. Currently, landmark detection is not available from Watson services. Therefore, other face recognition services/libraries (e.g. Google Vision API, APICloud.Me¹², Face++¹³, FaceRTMAPI¹⁴, etc.) could be used to retrieve those landmarks of a face. Calculating a distance between centres of eyes and corners of lips for each side of a face, we are able to conclude a presence of a stroke sign if difference between the distances is significant (higher than a threshold) (see Fig. 4). With a purpose to take into account possible personal initial asymmetry of a face, initial difference between the distances is applied to the threshold. To speed up a processing time and to reduce amount of unnecessary image features in classification models, all the snapped faces pass through pre-processing that cuts unnecessary parts of the images as well as transforms them into grayscale mode.

B. Speech

The logic of the App in the context of speech-based stroke indicators detection is shown in the Fig. 5. The goal of conversation on different topics is to collect the answers and store them as learning samples (members of clusters associated with the topics). Therefore, having set of topics with corresponding set of relevant to the topic questions and answers, we are able to train Watson Natural Language Classifier service to identify an intent (in our case - a topic) based on any further provided answer. If recognized intent differs from the topic that the asked question is associated with, it would indicate a possible irrelevance of the provided answer to the question and, as a result, will imply possible existence of “nonsense” problem. To increase the confidence level of the result, Classifier could be also trained base on domain/topic related keywords. This approach could be used in case when whole sentence (answer) is irrelevant/nonsense. Nonsense detection on the level of an individual word is trickier task, since it is quite difficult to detect whether certain particular word is relevant/irrelevant to the topic (could be considered as a part of future work).

With respect to the unobtrusive detection of “slurring” problem, App uses Watson Speech-to-Text transformation service. The service gives a possibility to see a confidence level of each word in the phrase, as well as array of acoustically similar alternatives for words with corresponding confidence levels. Based on this information and setting up a certain confidence level threshold for each word and for the whole sentence, we may detect a moment when service performs low quality transformation that might be caused by unclear pronunciation of a stroke affected user. Therefore, when difference between current confidence level and control confidence level (associated with “healthy” user) is bigger than a threshold, we may indicate a stroke effect. Under the confidence level we may consider either the lowest confidence level of one of the meaningful word (not articles or very short word) in a phrase, or some aggregated value (average or mean value of the confidences of all meaningful words in a phrase). For current version of the prototype we use aggregated mean values of the confidences in the “Friend” mode. Of course, such a result might be caused by other reasons as well (e.g. when person is drunk, under some medication affect, etc.). Anyway, this is a reasonable detection to activate the “Doctor” mode of App’s business logic - more precise detection of stroke probability.

In the “Doctor” mode, App uses more sophisticated method to calculate probability of a stroke. In this mode, App does not pretend to have a natural conversation anymore. In contrast, it starts to behave as a doctor who performs some tests with a purpose. With respect to the “slurring” problem, App asks user to repeat certain predefined phrase(s) that is usually used by doctors to recognize the problem. Such specific phrase usually contains special combination/sequence of sounds that become extremely difficult to pronounce for a stroke affected person (e.g. “Don’t cry over spilled milk”, “you can’t teach an old dog new tricks”, etc.). Since we are not using sound processing techniques directly, App uses the same Watson Speech-to-Text service and retrieves words’ confidence levels from the result. This time, to be more precise with stroke prediction, App compares the confidence levels against words’ “control” confidence levels collected for the same phrase(s) pronounced by user in “Setup” mode, when (s)he was healthy (under the normal, “healthy” condition). For this purpose, App asks to repeat special phrase(s) when the user runs the app for the first time (“Setup” mode of the App), and save the result of speech-to-text transformation for later comparison. This time App takes the biggest difference (delta) between the confidence levels of a meaningful word in a phrase of recently captured and “control” values to compare with a threshold, as the most careful extreme for the decision making.

Additionally, to tackle the problem of other possible reasons that might cause “slurring” problem (drunkenness, medication effect, etc.), App tries to recognize various contexts (e.g. party, restaurant/pub location, dental treatment, etc.) based on conversation content and through the possible use of other context provisioning channels (e.g. sensors, calendar, etc.) and asks user to repeat special phrase(s) to link the result with the context. Having a record of context related results of the same phrase transformation we may use corresponding measurements of the words’ confidence levels to make more

¹² <http://apicloud.me/apis/>

¹³ <https://www.faceplusplus.com/>

¹⁴ <http://api.animetrics.com/>

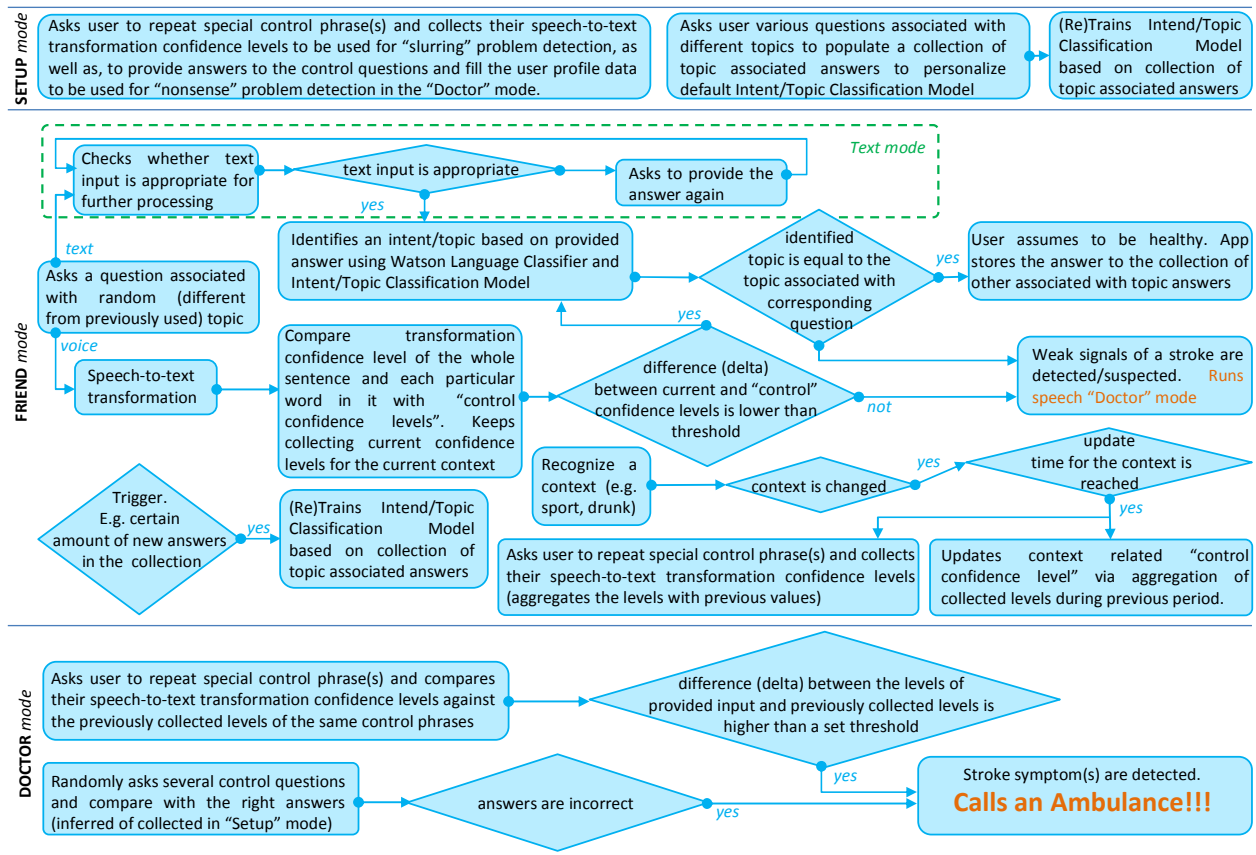


Figure 5. Speech-based stroke indicators detection logic.

correct stroke prediction with respect to the current context. Following the same approach we may keep a record of generalized (averaged) context related threshold to be used in the "friend" mode (unobtrusive weak stroke indicators recognition) instead of some general default threshold. To address the "nonsense" problem in the "Doctor" mode, App asks some well-known general facts (e.g. "What a season (month, year) is now?", "What country are you now?", etc.), and/or questions specific for a particular user (e.g. "How old are you?", "How many children do you have?", etc.), and/or asks to perform some simple calculations (e.g. "2+4", "8-3", etc.). To check a correctness of provided answers and decide about "nonsense" problem existence, some of the questions will require simple reasoning based on and/or comparison with "control" data collected in "Setup" mode. Since both speech problems might occur simultaneously, triggering of any of them in the "Friend" mode will lead to a more comprehensive check of both of them in the second, "Doctor" mode.

It is very important that App works under any conditions, it should keep working in noisy environment as well. For this purpose App facilitates voice based communication with text messages on the screen in the "voice mode" (to ensure that user understands App), as well as supports "text mode" where user enters inputs by typing the text messages. However, text based communication brings challenges to speech based problem detection. It becomes impossible to recognize "slurring" problem in the "text mode", since App does not use Speech-to-Text transformation. Also, we have to slightly change the "nonsense" problem detection logic. If voice input

transformation in the "voice mode" returns insufficient confidence level, App considers this as a "slurring" problem and does not processes the input to detect "nonsense" problem. Therefore we have to define the way we deal with inappropriate text input (kind of uncontrolled input of a set of symbols), since Natural Language Classifier service returns some (most close/relevant) topic for any input we provide. To tackle this problem and recognize an unappropriated text input, we may consider several approaches.

Making an assumption that "uncontrolled sequence of characters" would not be understood by Watson language processing tools, such input will be referred (classified) to many topics in the Watson Natural Language Classifier service model with pretty much the same confidence levels. Thus, if differences between confidence levels of top N classes/topics (input is classified to) are within certain threshold, then the input might be considered as an inappropriate, and App asks user to repeat an answer. Another approach is to introduce one extra class for Watson Natural Language Classifier service that supposes to cover all the inputs that are not relevant to any of predefined topics (classes) in the model. In this case, we have to generate relatively huge learning sample of possible inputs that will represent this extra class. As we can see, it might require a lot of work to collect such a sample. The third option (that might be the most reasonable) is to use Watson Language Translator service and translate the input sentence into one (or several) another, well supported by service, languages. Low confidence levels of translation could be interpreted as an evidence of an inappropriate input.

C. Arm

Being equipped with specific motion and muscle strength monitoring sensors on both user’s hands, App would be able to be unobtrusive as well in detection of an one-side arm weakness stroke signs. Unfortunately, in current version of the App, we have an assumption to use only smartphone and sensors that are available in it (e.g. accelerometer and gyroscope). With current restriction, App could not be that unobtrusive and will not have separate behavior for “Friend” and “Doctor” modes (will work only in “Doctor” mode). Thus, to check existence of an one-side arm weakness stroke signs, from time to time App asks user to perform certain physical manipulations with smartphone hold in one arm and to repeat the same for another arm as well.

Prior starting the test, App provides an animated guidance that describes a process of the test (see Fig. 6). However, users (who are well familiar with the process already) may skip the guide and proceed to the test immediately. Being guided by voice instructions, “beep” sound and vibration, App asks user to keep the hand for 5 seconds in three positions: down (along with the body), in front of (90 degrees to the previous position) and up (above a head) positions. As soon as user starts the test, App recognizes first stable position of the phone (movement speed is less than a set threshold) and associates this position with the “down” position one of the arms - a starting control point $P_1(x_1, y_1, z_1)$ in the three dimensional space. After 5 seconds of keeping the arm in the same position (radius of movement is less than a set threshold), facilitated with “beep” sound and vibration, App asks user to raise the hand to the “in front of” position and associates another detected stable point with second control point $P_2(x_2, y_2, z_2)$. And again, after 5 seconds of holding the arm in the recognized control point, App asks user to move the hand towards third “up” position. After the first arm, App asks user to change the hand and proceeds with another arm in a similar way.

We may conclude presence of a stroke symptom either through big asymmetry in capabilities of different arms and/or inability to hold/keep arm in the same position for some time interval (in our case 5 seconds) and/or inability to raise a hand to the requested position. Taking into account possible difference of physical capabilities of the user from some, so called “normal/usual” capabilities; we would take into consideration a difference (delta) between positions of currently measured control points and previously collected “control sample” of the points rather than to compare them against of some general fixed positions. Therefore, additionally to main “Doctor” mode, App also has “Setup” mode when “control sample” of the control points are collected (see Fig. 7). Functionally App does the same logic in both modes. However, “Setup” mode is meant to be run under third party supervision (e.g. doctor in the hospital) to verify a “healthy” condition of the user during the “control samples” collection. Collection of new “control sample” could be done from time to time to keep more recent updated values. It could be done in a hospital under doctor supervision during one of the user’s visits or somewhere else under control of some other authorized third party (e.g. family members). In any case App should support third party verification mechanism to be sure that “control samples” were taken under “healthy” state of the user.

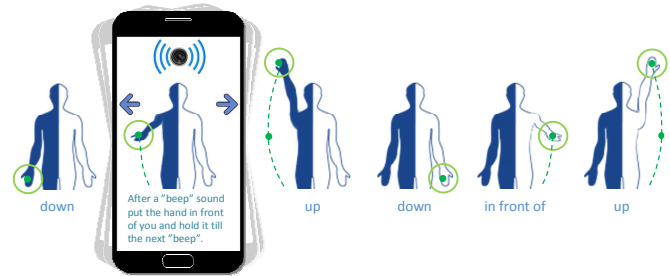


Figure 6. Animated guide.

Thus, comparing the values of currently measured control points against “control sample”, App is able to infer a level of inability to raise a hand to certain level and to keep it at the same level for certain time interval, as well as, to calculate a difference between the measurements with respect to different arms. Having the “delta” bigger than certain threshold, App concludes presents of stroke symptom. Moreover, having a “control sample” of the measurements taken in “Setup” mode and a log of previously taken measurements of “healthy” user, App increases a confidence of made decision by comparing latest measurements (of potentially affected by stroke user) against aggregated values of “control samples” of a “healthy” user. Such historical log of measurements will also allow us to avoid incorrect decisions in case if certain personal physical capabilities of the user were changed (developed or worsened) due to some other reasons.

D. Integrated logic

Described above methods for stroke indicators detection are parts of an integrated logic of the App (Fig. 8). As soon as App is activated, it runs at background and initiates interaction with user in the “Friend” mode according to interaction settings (time intervals, conditions, etc.). App could be activated manually with a purpose to run “Setup” mode (to change the settings, and/or to update control measurements and populate learning samples associated with “healthy” user), or when user feels unconfident about his/her health conditions and would like to run “Doctor” mode to be tested immediately, or even make a call to an ambulance via “one button press” in the App.

Weak “face” stroke signs could be detected during a communication between App and user; and corresponding detection logic is run any time the App is activated in any mode. Thus, App has only two triggers that follow “Smart Interval” logic and activate “speech” and “arm” stroke detection flows accordingly.

Default activation of “speech” flow could be scheduled by setting a “wait time” interval. As soon as wait time is over, App calls to the user and pop-up widget appears on the phone screen (facilitated with ringtone and vibration). From now, user has two options: either to start communication, or reject/postponed the call. If call is rejected or is not answered, App repeats call for several times with incrementally reduced time intervals. If user is not reachable within certain attempts, App sends warning notification to a third party (e.g. doctor/nurse, relatives, etc.). The same approach is applied even if user accepts the call, but turn-off the App without conversation. Thus, next activation time is updated only when App completes test(s) in “Friend” mode, or “Doctor” mode, if

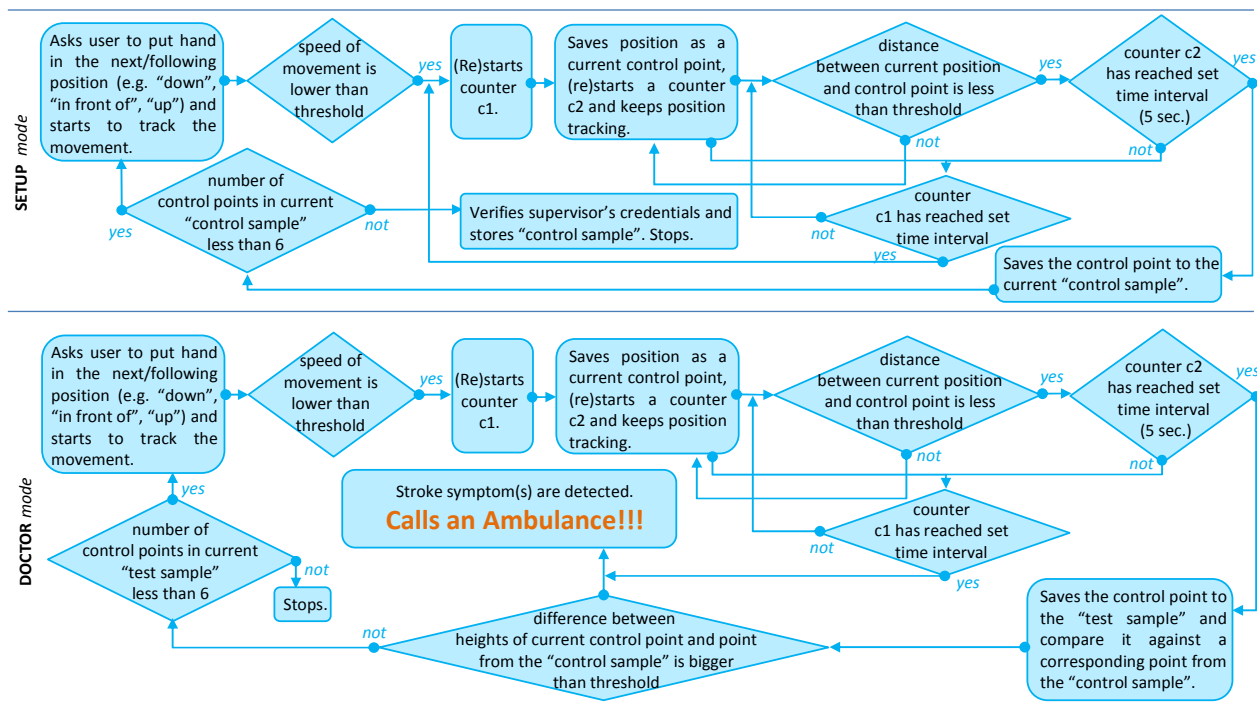


Figure 7. Arm weakness based stroke indicators detection logic.

it is triggered. “Smart Interval” logic of the trigger assumes that the “wait time” interval is not fixed, rather dynamically updated depending on various conditions and actions to keep App unobtrusive as much as possible. For example, if App has been activated by user manually, and health condition of the used has been tested, next activation time will be rescheduled with a purpose do not bother user too often.

“Arm” stroke detection flow has a separate activation trigger. However, it could be set for simultaneous activation together with “speech” flow, or set with different “wait time” interval (should be agreed with doctor), since it takes more time from user to complete the test in comparison to “speech” flow. At the same time, just because of mentioned difference, it might be reasonable to activate “speech” flow every time when user done with the test in “arm” flow. In this case, following “Smart Interval” logic, next activation time for “speech” flow will be rescheduled. Similarly, whenever weak stroke signs are detected in “speech” or “face” flows, and corresponding “Doctor” mode is activated, all three types of stroke indicators should be checked starting with the flow where weak signs were detected. Such unscheduled activation of the flows leads to rescheduling of the next activation times for the triggers.

As soon as App detects a stroke indication in “Doctor” mode, it initiates a process of user assistance (“Call an Ambulance” in the figure). Normally (without the App), a person, who has recognized a stroke symptoms him/her-self; or a person, who noticed that someone else is under a stroke attack, calls an ambulance and explains a situation. The App not only detects symptoms of a stroke automatically, but also makes the call instead of a human and in voice explains a reason of a call as well as shortly provides necessary information such as: name of the user, location, etc. Additionally, App gives a possibility for user to continue a conversation with an ambulance call-center representative to

insure realness of the case. In case user is incapable to speak due to a stroke effect, person in a call-center will hear unclear speech as extra evidence/proof. The same is done when user itself recognizes a symptom(s) of a stroke, but, due to affected speech, cannot able to make a normal call to an ambulance and use “one button press” option to make a machine-driven call instead. At the same time when App makes a decision about existence of a stroke symptom(s) and calls to an ambulance, it can provide useful information that will facilitate a final decision making of a person in charge. It could be samples of speech/face/arm based stroke indicators collected and used by the App for the made decision. Also, in case of a call initiated by the user via “one button press” call function of the App, couple snapshots of the user’s face could be taken by the App (while user does it). Thus, for the purpose to access such supportive data that may facilitate further decision making, we may consider a possibility to equip a call-center of an ambulance with a web application through which a related to the call data will be easily accessed.

IV. PROTOTYPE IMPLEMENTATION

This section presents the aspects of prototype implementation. Solution uses IBM Bluemix Continuous Delivery which automates the build, test and delivery of the applications. Stroke App was developed based on Node.js Cloudant DB Web Starter Boilerplate which contains Node.js server and IBM Cloudant NoSQL database. This boilerplate was heavily modified for the development purposes. Initially backend was configured to be appropriate for the use in this development process. There was no need for changes in the IBM Cloudant DB. Also, it should be noted that Continuous Delivery works at this moment only in US South region. Stroke App is connected to Availability Monitor, Natural Language Classifier and Speech to Text services in IBM Bluemix platform. Node.js

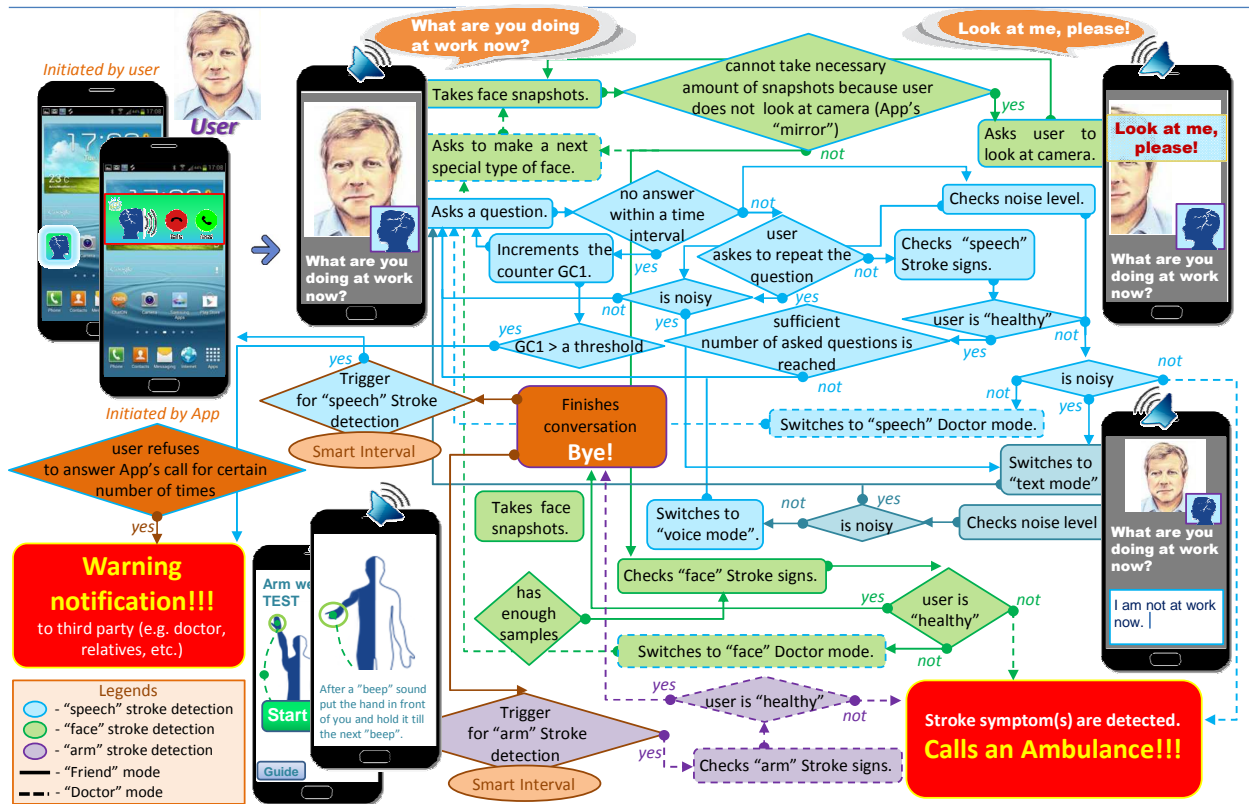


Figure 8. Integrated Logic of the App.

provides backend functionality to our application which can be used as web application via web frontend or via Android application with http requests. Different services are used with Node.js backend and the credentials for the services are needed to connect to these services. Service credentials were obtained as JSON file for the local use as environmental variables inside the development environment when boilerplate was initiated and in production environment service credentials used were present in deployment server as environmental variables.

With respect to face-based detection approach, we have tried both: deep neural network image classification (with IBM Watson Visual Recognition service) and Landmarks based approaches. The drawback of image classification based approach is required a big enough training dataset of images, when we are talking about user-personalized classification model. In real life it might be almost impossible to collect sufficient training set of user's images that represent a stroke affected face of a user. Also, generalized classification model trained on images of different people does not return high enough confidence level for individual patient. Therefore, the optimal choice so far is landmark based approach with image splitting and mirroring.

After App gets the face photo (Fig. 9), App crops it reducing unimportant parts and background. Older version of Google Vision API works offline, which is beneficial for our purposes, since user might not be connected to the internet constantly. It allows detecting faces, counting amount of them in a frame and detecting landmarks. We also do dummy check for eye opened probability – if they are not equally opened, this is the fastest sign of face asymmetry. The Face API provides

important measurement of Euler Y and Euler Z (but not Euler X) angles that characterize a face's orientation and help us to normalize the image. However, we have noticed a problem straightforward use of API. Google API perfectly detects landmarks, but (similarly to other landmark detectors we checked) use assumption that face is symmetrical. It does not provide raw detected location, but try to correct position of landmarks to bring them closely to symmetrical positions, which is not acceptable for our case. Therefore, we used landmark in combination with face splitting and mirroring. In this case we get two symmetrical images out of original image. And, in case of significant asymmetry of original image, difference in distances between landmarks associated with eye and lips corner on two generated images will be notable. We found this approach the most reliable for our purpose.

Regarding speech-based detection approach, we did not face any problems and our tests have shown pretty good results

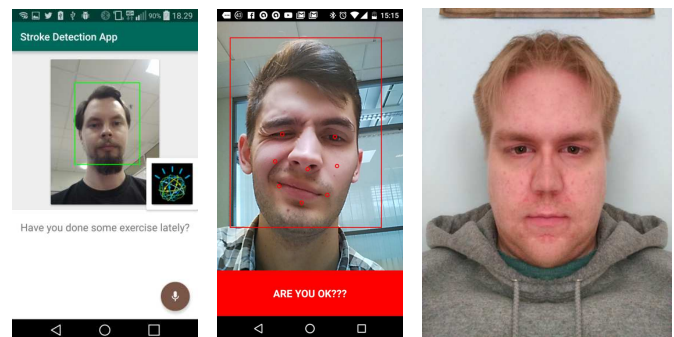


Figure 9. Face and Landmarks detection, and example of flipped image.

To train Watson Natural Language Classifier service to recognize intents from the conversation this user, we have prepared corresponding questions and set of possible answers for such topics like: Family and Friends, Work, Physical activity and Hobbies, Holidays/Travelling, Food, Music/movies/TV-series, Weather, Animals, etc.

To track a location of the smartphone in the space and collect coordinates of control points for arm-based detection, we use the acceleration sensor's data. With the acceleration data and the time value, we can calculate the velocity and the position. However, we have faced some problems with this approach. First, we have to take the current rotation of the phone into account. Different rotation of the phone causes different values from accelerometer. Due to limited time, we did not find proper algorithm for correct calculation. Therefore, our current solution has corresponding limitation - for every tests we do, we have to hold the phone at the same angle (facing upward).

V. VALUE OF THE SOLUTION

The benefit and actual value for the users of offered solution (people under the risk of stroke attack and their family members) is obvious. Lives and health are priceless and absolutely beneficial for society. Talking about economic benefits, reducing amount of delayed cases, we are able to save expenses for those who covers costs for expensive advanced treatment and rehabilitation. The light treatment of a stroke (thrombolytic therapy) usually can be applied in local hospitals, while more advanced and more expensive treatment could be available only from special medical centers that might be difficult to reach in time. For example, a light treatment can be given in Jyväskylä at the Central Finland Central Hospital for 3-4 hours after the symptoms have started. In more difficult cases the patient needs to be transferred from Jyväskylä to Kuopio University Hospital where the vein is opened. According to the statistics [5], there are appr. 14 600 stroke incidents yearly in Finland and the stroke happens again to 2 500 people within the first year. Thus, there are 17 100 stroke cases in total. With respect to [6], cost of the patient having first stroke is 20 400e in a year, and lifetime cost of the stroke patient is 86 300e. However, present solution is able to reduce amount of delayed cases. Therefore, to estimate the potential benefits that it might bring, we have to address the difference between the costs of light&cheap and difficult&expensive treatment. According to hospital's cost chart, such difference is about 1571e. Taking into account amount of yearly delayed patients (250 patients in Central Finland Central Hospital and 7300 patients in Finland), the direct benefits are 392 750e and 11 468 300e per year respectively. At the same time, there are might be some indirect costs that additionally cover more advanced rehabilitation procedure as well as lifelong support of people with disabilities caused by treatment delay. Taking into account that amount of strokes happened in the world in 1000 times more than in Finland (appr. 17 million), potential savings worldwide could be billions of euro yearly.

VI. RELATER WORKS

There are some products (mobile applications) related to stroke are available on market already. All of them are mainly

designed for human as a tutorial to make him/her aware of a stroke, its symptoms, ways of treatment and rehabilitation. They guide person via human readable stroke assessment tutorial and help to recognize the symptoms that another person might have. There are some examples: F.A.S.T. stroke app (Spot a Stroke F.A.S.T.)¹⁵ is offered by American Heart Association/American Stroke Association, Stroke Awareness 101¹⁶, etc.. Other apps assist user to estimate a risk of a stroke (Stroke Riskometer¹⁷) and help to choose appropriate antithrombotic therapy for stroke prevention (AF-STROKE¹⁸); to explain strokes, available treatments and the hospital care process (Stroke Patient¹⁹); or even provide rehab assessments to help with post-stroke recovery and prevent its recurrence (9zest Stroke Rehab²⁰). However, we did not find any apps and systems available on a market that are capable itself automatically detect stroke symptoms.

VII. CONCLUSIONS

This paper presents the approach we have used for prototype implementation of Stroke Cognitive Medical Assistant - smart supportive system for people under high risk of stroke attack that is capable to automatically recognize stroke symptoms at their earlier stages, and prevent a significant delay in treatment, minimizing possible harmful consequences. In this works we have studied different approaches and techniques to achieve functional requirements of the solution. As a future work, we are planning to focus on improvement of the arm-based detection technique and find appropriate algorithm to achieve more sufficient performance. Also, we plan to not only use smartphone sensors, but also use smartwatches and other wearable devices to receive necessary data for arm-based detection.

ACKNOWLEDGMENT

The research is done in the "VALUE FROM HEALTH DATA WITH COGNITIVE COMPUTING" project led by Faculty of Information Technology (University of Jyväskylä) in collaboration with IBM, Central Finland Central Hospital and other medical/social care organizations. Authors are thankful to other project team members for fruitful collaboration and useful discussions, especially to Pirjo Mustonen (cardiology expert at Central Finland Central Hospital) for consultancy with medical matters.

REFERENCES

- [1] T. Berners-Lee, J. Hendler and O. Lassila, The Semantic Web, Scientific American 284(5), pp.34-43.

¹⁵ <http://news.heart.org/f-a-s-t-stroke-app-helps-save-a-life/>,
https://play.google.com/store/apps/details?id=au.com.n6.stroke_us_android&feature=search_result#?t=W10

¹⁶ https://play.google.com/store/apps/details?id=com.app_mikeaa.layout

¹⁷ <https://www.healthtap.com/apps/38635>

¹⁸ <https://play.google.com/store/apps/details?id=com.cardioneurology.af>

¹⁹ <https://play.google.com/store/apps/details?id=com.neurorecovery.strokepatient.android>

²⁰ <https://play.google.com/store/apps/details?id=com.ninezest.stroke>

- [2] T. Heath and C. Bizer, *Linked Data: Evolving the Web into a Global Data Space* (1st edition). Synthesis Lectures on the Semantic Web: Theory and Technology, 1:1, 1-136. Morgan & Claypool. 2011.
- [3] V. L. Feigin, M. H. Forouzanfar, R. Krishnamurthi, G. A. Mensah, M. Connor, D. A. Bennett, ... and C. Murray, "Global and regional burden of stroke during 1990–2010: findings from the Global Burden of Disease Study 2010". *The Lancet*, 383(9913), 2014, pp. 245-255.
- [4] RU. Kothari, A. Pancioli, T. Liu, T. Brott and J. Broderick, "Cincinnati Prehospital Stroke Scale: repro-ducibility and validity." *Ann Emerg Med* 1999 Apr;33(4):373-8
- [5] Hippocampal, Cerebral Palsy (AVH) in figures. (2013). Subsequent 7.8.2017. https://www.aivoliitto.fi/files/1091/avh_lukuina2013_web.pdf
- [6] A. Meretoja, "PERFECT stroke: PERFORMANCE, Effectiveness, and Costs of Treatment episodes in stroke." Dissertation, University of Helsinki, Medical Faculty, 2011. <https://helda.helsinki.fi/bitstream/handle/10138/26460/perfects.pdf>

ABOUT THE AUTHORS



Oleksiy Khriyenko (1981) obtained Engineer's degree in Computer Science (Intelligent Decision Support Systems) in 2003 from the Kharkov National University of Radioelectronics, Ukraine. Later, Oleksiy Khriyenko obtained a Master's degree in Mobile Computing from MIT department (University of Jyväskylä, Finland). Since 2008 he is Ph.D. from the same department. His

research interests include: Artificial Intelligence, Deep Learning and Cognitive Computing, Semantic Web and knowledge engineering, multi-agent systems, Web of Things and ubiquitous services, context-sensitive adaptive environments, etc. Currently, Oleksiy Khriyenko does research, lecturing and is involved in management of international master programs (WISE, COIN) at IT faculty, University of Jyväskylä. (<http://users.jyu.fi/~olkhriye>)



Konsta Rönkkö is an Information Systems Science student graduating in 2018. He has worked in various projects as a Research Assistant at University of Jyväskylä's IT Faculty since 2015, most recently as a Software Development Lead in Value from Public Health Data with Cognitive Computing and Watson Health Cloud Finland projects. Konsta Rönkkö's research interests focuses

mostly on Natural Language Understanding and Processing, nowadays working as a Technical Specialist at IBM Finland.



Vitalii Tsybulko (1995) graduated from Kharkiv National University of Radioelectronics, Ukraine, where he gained a Master's degree in Computer Science, focusing on Artificial Intelligence Systems. Currently, Vitalii is obtaining Master's degree in the Faculty of Information Technology, University of Jyväskylä. As research assistant, was involved in projects "Value from

Public Health Data with Cognitive Computing" and "Health Cloud" at the University of Jyväskylä, where contributed mostly to the visual recognition and architecture design. Among scientific interests, Vitalii has neural networks, audio and visual recognition.



Kalle Piik is studying as an undergraduate student at University of Jyväskylä, Information Systems Sciences and was involved as a research assistant into the "Value from Public Health Data with Cognitive Computing" project at University of Jyväskylä in 2017. Kalle's research interests covers distributed systems, cloud computing and cognitive computing. Currently, he is working as an IT Specialist at

the Social Insurance Institution of Finland focusing at distributed processing frameworks (incl. Hadoop and Spark).



Duc Pham Minh Le (1993) graduated with a Bachelor of Information and Communication Technology (ICT) in University of Science and Technology Hanoi in 2014. At this moment, he is pursuing his Master's degree in Web Intelligence and Service Engineering at the University of Jyväskylä, Finland. In 2017, he worked as a research assistant in the "Value from public health data with cognitive

computing" and "Health cloud" initiatives at the University and is now a software developer in the incubator team for a robotic project at Solteq plc, Finland.

Tommi Riipinen currently is working as a project team leader at IT Faculty, University of Jyväskylä, overseeing AI prototype development and robotics research groups in the project "Watson Health Cloud Finland". His research interests include natural language processing, machine learning and service robotics.