# Helping People with Language Learning Disabilities Using Native Mobile Voice Recognition — Exploring Its Limits and Advantages

C. Alonso, T. Read, and J. J. Astrain

***Abstract*—This article presents an intelligent system for people with language learning disabilities called MarLuc. MarLuc aims to improve people's skills in their native language rather than a second language. The system was born as a tiny Computer Assisted Language Learning (CALL) web-based application targeted to improve pronunciation. Later it became a Mobile Assisted Language Learning (MALL) application and has therefore incorporated some powerful native resources such as voice recognition present on mobile devices. Working with personal mobile devices has brought new exciting possibilities, including but not limited to easy access to native voice recognition and the utilization of bots. In this study, we will show the advantages and some limitations of using native Siri or Android based voice recognition.**

***Index Terms*—MarLuc, computer assisted language learning, mobile assisted language learning, language learning disabilities, hybrid mobile applications.**

## I. Introduction

Over the decades CALL has helped people improve their second language competences. However, for native speakers who are experiencing difficulties with regards to speech, no such equivalent progress can be identified in the literature. This is somewhat surprising since it is estimated that between 6 and 8 million people in the United States have some form of language impairment [1]. The term language learning disorder or impairment is hard to define as there is not a clear classification of the problems underlying these disorders and the existing ones vary depending upon the academic discipline where they are analyzed. Even when one specific disorder is identified, there may be several forms and different symptomology (e.g., people that share the same disorder could suffer comprehension problems but not production ones, and others the opposite). For certain disorders, noting that nonlinguistic, more cognitive difficulties (such as logical reasoning) may also play a part in the language disorder, can greatly complicate the task of any therapist [2]. Finally, in the case of children, even defining the term 'child language disorders' and what it implies, is still under debate [3].

Comparative studies about the results of treatments and therapies on people with language disorders has never been an easy thing to do [4]. It is hard to find a population with the same disorder and degree of impairment, similar social environment, etc. Fortunately, several years of research on practical treatment approaches have led to the generation of valid results on certain common patterns seen in different disorders [5], [6]. The work presented in this article represents an attempt to complement the standard face-to-face therapeutic treatment provided by speech therapists, and other related professionals, to potentiate learning beyond just undertaking the exercises provided by the therapist between sessions.

## II. Design Foundations

This work started with the design of a CALL system intended for use on desktop and laptop computers. Its initial features where based on a set of simple exercises proposed by therapists. One of the design foundations of the project was that it should be a public web-based service to serve the highest number of people possible. Nevertheless, by the time the application had become functional the growing computing capabilities of mobile devices made it necessary to revise this decision. It was clear that mobile devices in form of "smartphones" and tablets soon would become the preferred way of accessing computing resources, especially for children. Modern smartphones are much more than just communication devices. Their connection to Internet makes it possible to use these devices for learning activities, giving rise to possibilities for 'anytime, anywhere' activities. The application of such devices for language learning has been amply documented in the literature [7]-[9] giving rise to the term Mobile Assisted Language Learning (MALL). Their reduced cost makes them even more attractive to students, and has been noted in the literature, where their use motivates students to learn [10]. For these reasons it was decided to extend the design of the system to convert it into a MALL app for smartphones and tablets, which is what is detailed in the rest of this article.

Although changing a desktop system to a mobile application was clearly the right option, rebuilding the whole development from scratch would not have been feasible. At this technical crossroads, finding the right path was not easy. In the mobile field, there were several possible solutions for application development [11]. Unfortunately, they were actually incompatible among themselves, mostly depending on choosing between the two main development environments; for Android or for iOS [12]. Fortunately, there

Manuscript received December 19, 2019; revised March 10, 2020. This work was supported in part by the Government of Navarre (Spain) under research Grant 0011-1365-2018-000157 (TuSpeaking).

C. Alonso and J. J. Astrain are with the Department of Statistics, Computer Science and Mathematics, Public University of Navarre (UPNA), 31006 Pamplona, Spain (e-mail: losalo@unavarra.es, josej.astrain@unavarra.es).

T. Read is with the Department of Languages and Computer Systems, National University of Distance Education (UNED), 28040 Madrid, Spain (e-mail: tread@lsi.uned.es).

doi: 10.18178/ijiet.2020.10.8.1429

was an option in which we could maintain the core of the application while changing only the front end, enabling the application on mobile devices. Using hybrid application frameworks, development is based on HTML5 [13], CSS and JavaScript. Hybrid frameworks such as Apache Cordova [14] provide a common interface to the internals and graphical possibilities of each type of device while developing in HTML5, so the same code can run on any of them. The developer's code and the interface layer are packaged together, and the result is as if it were an ordinary native app. From the user's point of view, it is like another native app downloadable from the corresponding app stores. For programmers it offers the possibility of building the same code for Android or iOS and also to reuse code developed for the web-based applications, as we have done. The app's main menu can be seen in Fig. 2.

After some time spent adapting the app to this new environment, we finally got the app working on mobile devices, not only with the same features as before, but even with more and powerful functionality.

## III. System Description

### A. System Architecture

The system was conceived to be easily maintained and, in a way, so that new words, sounds or data can be added without the need to build a new version of the app. Many of the menus and their options are also stored externally in a content manager so can be updated or changed without changing the core of the application. The mobile core or the mobile presentation layer was developed using the Ionic Framework [15], [16]. This framework is one of the most successful tools used to develop hybrid applications [17], [18].

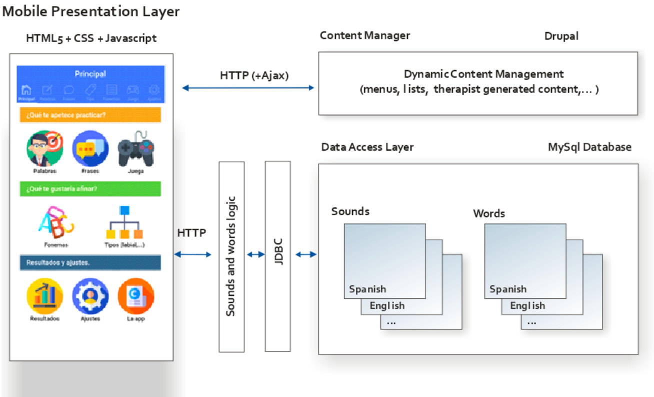


Fig. 1. System architecture of MarLuc.

One important addition from the desktop version in the new architecture, is the use of a Content Manager System (CMS). Using a content manager enables the use of a very tiny mobile-dependent layer, thereby delegating many interface and content management aspects to the content server. Furthermore, the computationally intensive calculations undertaken as part of the content adaptation process can be undertaken on the project server or even cached, thus the application could be more powerful than it could ever be as a standalone MALL app. The content manager also serves as a user-friendly editor for new content or data that can be incorporated directly from therapists. Although it required the addition of a new layer to the application schema, the layer will drastically reduce the maintenance of the system and greatly adds to its longevity. Drupal was selected as the engine for CMS.

The words and sounds database is held in a small web application using Tomcat. In fact, the sounds and words manager is the same piece of software that was in the desktop version. It is designed to be multilingual and to manage words and sounds in different languages, although in the phase of the project we are centering on the Spanish version only. The architecture of the system is presented in Fig. 1.

### B. Overview of the System

The system was originally developed for use with a web-based interface. However, with the extended use of mobile devices, smartphones and tablets, it was decided that mobile deployment would be more appropriate for learning. It has two parts: a MALL app and a server-side application. The app handles local interaction with the user but also benefits from the processing power of external servers as we will describe later for content and support of recognition system. In its current form, MarLuc helps the user practice vocabulary, containing more than 8,000 words. The sound of the words can be heard at different speeds to capture differences of sound within a word. This can help people who, for some reason, have trouble perceiving the details of the sound patterns of words pronounced at normal speed. Thus, the sound of the words can be heard at 100%, 50% or 33% of its actual pronunciation speed. These words are recorded by a person and are not a machine text-to-voice generated sound. We considered it important to work with a real person's voice to get the actual sound of the word right for learning. The app also incorporates a voice recognition option, so users can practice words and can have their production checked against it. As will be detailed later, the voice recognition system is helped by a bot system in the background to gain accuracy. The app has also been gamified and includes a small game that enables user to pair up heard words with their equivalent images. The results of any of the exercises done by the user can be sent to the therapist, so the results can be checked and assist in planning the next steps of therapy accordingly. The application is intended to be an additional aid in speech therapy, especially at home, where an adult supervises its use

### C. List of Features of the System

These are the main features of the system:

- The application contains the sounds of more than 8,000 human-recorded words to practice pronunciation.
- The sound of the words can be heard at different speeds to capture the differences of sound within a word. This can help people who, for some reason, have trouble perceiving the details of a sound pattern at the normal speed at which we pronounce the words.
- It uses mobile or tablet native voice recognition, so it can verify whether the user pronounces correctly the words presented.
- Word or word by type can be selected for practice; alveolar, bilabial, etc.

- A similar selection can be made for each phoneme /a/, /b/, …
- A simple game is included with sound and images to practice vocabulary
- The application allows the results of the practice to be logged, and if desired, sent automatically to a therapist, so s/he can be aware of developments at home. The therapist then can plan more precisely the next visit and the new steps required in therapy.
- As well as the automatic data logging, the app also includes a feedback form to help users provide comments and make requests for improvements.

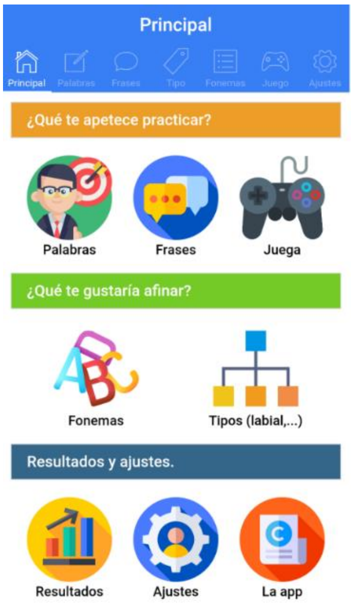


Fig. 2. MarLuc main menu.

## IV. Interacting with Mobile Native Speech Recognition

Moving to a mobile framework offered the possibility of accessing the speech recognition features available on mobiles and tablets. As mentioned before, we decided to employ hybrid mobile frameworks based on HTML5 instead of using native coding in iOS or Android as development platform. This option let us use much of the code already done for the desktop version and share the same code for iOS and Android. However, this choice also had some drawbacks, as we cannot reach native resources such as the speech recognizer present on the mobile device directly. There is an interesting web standard, called Web Speech Recognition API [19], that represents a bridge between the native speech recognition and HTML5. The API sets the interface between HTML (and Javascript) and the native recognition system that could be present on the device.

Although the Web Speech Recognition API in HTML5 is quite advanced and actually present in some desktop browsers [20] it is not the case in the mobile world. Hybrid applications on mobiles rely on internal browser engines (named WebView). These mobile engines still do not implement natively the Web Speech Recognition API. In the meantime, while this standard is fully transported to the mobile devices, we can use third party options or middleware to emulate the interface and access similar features offered in the standard Web Speech Recognition API. Using the middleware we can get speech recognition interaction accessing Android assistant or Siri using HTML5 and javascript.

### A. Using Speech Recognition Present on Android and iOS Devices

For voice or speech recognition in Android and iOS, we rely on its native speech recognition system. Using the same framework and middleware [21] in both platforms to access it, once the speech recognition finishes, the app receives a text from the recognition system together with a set of possible alternatives representing the words the user has supposedly said. Also, each alternative has a confidence value, ranging from 0 to 1, where 1 is the highest confidence value.).

### B. The Importance of Context

One of the first things that becomes apparent when using an automated speech recognition system is the importance of the context. It something we can easily experiment each time we use a recognition system on our mobile devices. Context adds a lot of information about what is probably being said and eases the path to a correct answer. Context could be provided, for example, by the surrounding words or the situation in which sentences are said. Extracting information from the surrounding words is widely used. For example, if the system has trouble recognizing ‘parking’ or ‘barking’ as the last word in a sentence, then context can be used to reject “the dog is parking”, selecting “the dog is barking” as the correct one. Hence, a recognition system tends to provide logical or meaningful answers where there is not a clear confidence on the recognized words. In the same way that surrounding words can be used to reduce the correct possibilities, the situation can do same. Conversational interfaces or chatbots (or just bots) like Dialogflow [22], Wit.Ai [23] or LUIS [24] operate on the sub-languages present in concrete situations or predefined contexts as assistants of fast food sites, stores or travel agencies. The possibilities of what is going to be said is somehow predetermined and follows a predictable flow. There are many examples about how these situations can be handled by a bot [25]-[27].

### C. Benefits and Limitations of Context in Speech Recognition in Helping People with Language Learning Issues

The main aim of MarLuc is to help improve pronunciation for those people experiencing difficulties in their mother tongue. Hence, it focuses on practicing words and sentences and providing feedback about the quality of the results. For the first part of the application, recognizing just words proved to be a very challenging task. The recognition of just one word cannot be made easier by the information of the surrounding words nor does it have any additional information from the situation in which it is being said. Therefore, the system is less able to identify possible results.

Considering that lack of context could impact in the quality of our recognition system, we undertook some tests of its accuracy without considering any context. For this purpose, we made MarLuc recognize its own database of words. Users have the option to listen to any of the 8.000 words that it stores before practicing their pronunciation themselves. In the tests, we modified MarLuc, attaching its output (the stored voice) to the input (the speech recognition) in a loop. That way, we tested all stored words against the speech recognizer of the device.

As expected, the results showed that without any context, about 13-25% of the times the recognition system did not identify the word correctly. The tests also showed that in our case native recognition system on iOS performed slightly better (87.66%), than its Android counterpart (75.96%), as illustrated in Fig. 3.

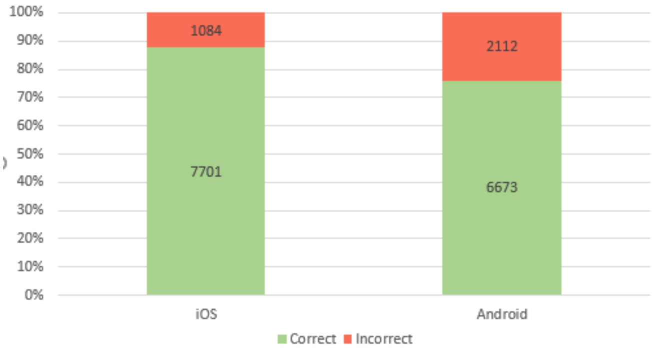


Fig. 3.Comparision of self-test of Marluc words against internal speech recognizer.

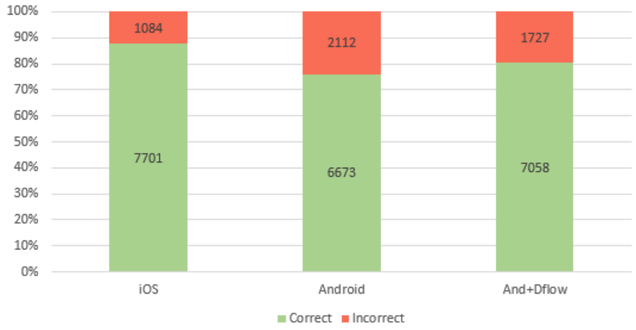


Fig. 4. Adding context to Android version improves recognizer results.

The next step was to find a way to introduce context into our system and test if we really could benefit from it when trying to recognize a single word. The Spanish language has around 100,000 words [28] if we consider the original Spanish words together with the addition of the Latin American ones. Roughly speaking, speech recognition must work using a set of 100,000 words when recognizing Spanish. A simple way to add context for us was to limit the set of the admitted results to the 8.000 stored words in the app. That way the speech recognition system could rule out the words that are not present in our set of words, thereby reducing the result alternatives, and increasing the possibilities of identifying a correct word. We use Dialogflow to provide context for our application. Dialogflow is a framework for building bots that could be used as assistants in certain situations. We made a simple bot aimed to recognize a word from our database of 8.000. We connect Dialog Flow to MarLuc in the android version to assist it in voice recognition using the Cordova connector for Dialogflow and audio capture plugins [29], [30]. Results show an improvement of 4.38% in the recognition of the same words, raising the success in recognition from 75.96% to 80.34%, as depicted in Fig. 4.

### *D. Limitations of Native Speech Recognition on Mobile Devices at Speech Therapy*

The main limitation of using mobile devices and their speech recognition systems in a speech therapy app ironically comes from their 'perfection'. This is due to the fact that they are primary designed for people without speech impairments. Thus, the speech recognizer's results are always in the domain of possible and valid words and sentences. In the worst case, they could provide a not very meaningful sentence as a result, but all words inside it would be correct. This behavior can be easily observed in the recognition of just one word. For example, if we have problems with phonemes 'b' and 'p', and we actually say "beace" instead of 'peace', the system would probably rise 'peace' as the result, as it could be the best right word that matches what it has received. In summary, even if we said something wrong, the speech recognition system would probably identify what we are trying to say.

While, it can provide a high quality fault-tolerant behavior for noisy places and dealing with bad sound quality, it can actually compromise the capabilities of diagnosis and correction in speech therapy. I.e., using the native speech recognition as is, will not show us the incorrect phonemes pronounced.

### *E. Can We Detect the Wrong Phonemes Using Native Speech Recognition with the Aid of External Systems?*

Systems like CMU Sphinx [31] work at phoneme level and can be capable of detecting 'wrong' words. These systems work with a database of words described by phonemes. They have databases for each language, mainly dictionaries of words defined by their phonemes. It is possible to extend their dictionaries with new words, inserting their corresponding phoneme patterns. That way it could be possible to include incorrect variations of correct words, like the previously 'beace' for 'peace'. Therefore, the incorrect word will be recognized and the problem at speech could be detected.

However, the task of including all possible 'wrong' words in speech by phonemes could be a significant challenge, if even feasible. Nevertheless, it is possible to find a compromise to limit the set of the wrong words to be detected. This can be done using the most common wrong variations detected in speech therapy. Even then, more than one phoneme can be wrong or displaced.

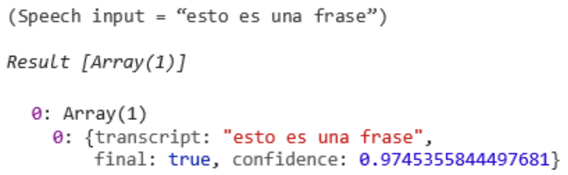
### *F. Improving the Detection of Speech Problems*

Although we have seen that there are some limitations finding wrong phonemes, and adding more accurate methods to find them could be a difficult task, we can still use some of the information provided during the speech recognition process to find a way to track what is going wrong in speech.

When called, the speech recognition process returns an array of possible results or alternatives. Each result has a confidence value. The higher the value of confidence is, the better the match. Confidence values range from 0 to 1; 0.95 indicates the system is quite sure about the word or sentence said and 0.65 would be a doubtful match. In practice, when the speech recognition is quite certain about what has been said, it returns a shorter array of results (or just one item), and the items inside it have high confidence values. Conversely, when the speech recognizer is not so sure about what should be the correct answer, the array of results is longer, and their confidence values are lower.

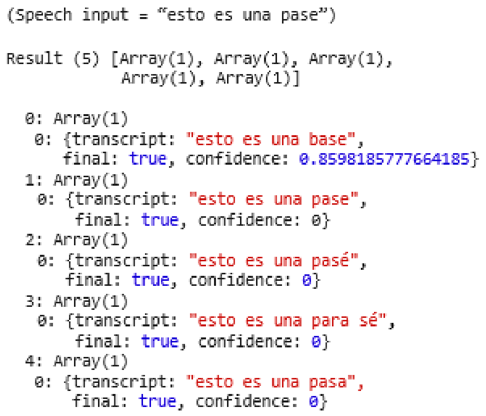
Therefore, depending on the size of the results array and its confidence values, we can indirectly detect there could some speech problems. Obviously, we cannot forget that there could be other circumstances like a noisy environment, bad sound quality, etc., that could degrade speech recognition results and not be due to problems of the speaker. In our case, as our application is intended to help in speech therapy, we assume it is going to be used in a place where speech recognition is not going to be compromised by the environment.

To prove this point, we made some tests with different sentences. We run them mainly on the Android version. First, we tried with correct and meaningful sentences. For example, one test sentence was "*esto es una frase*" ("this is a phrase"). The speech recognizer has no trouble in finding a match and then returning a single high confidence result, as shown in Fig. 5.



```
(Speech input = “esto es una frase”)

Result [Array(1)]

   0: Array(1)
     0: {transcript: "esto es una frase",
         final: true, confidence: 0.9745355844497681}
```

Fig. 5. Result in a correct and meaningful sentence.



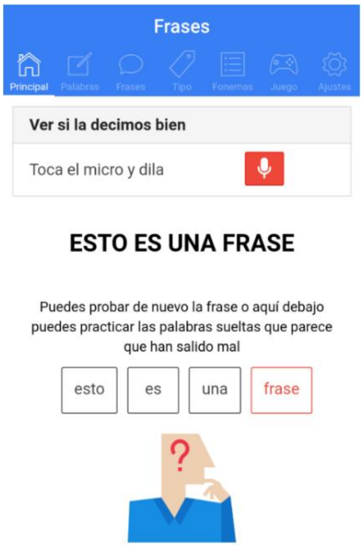
```
(Speech input = “esto es una pase”)

Result (5) [Array(1), Array(1), Array(1),
            Array(1), Array(1)]

  0: Array(1)
   0: {transcript: "esto es una base",
      final: true, confidence: 0.8598185777664185}
  1: Array(1)
   0: {transcript: "esto es una pase",
       final: true, confidence: 0}
  2: Array(1)
   0: {transcript: "esto es una pasé",
      final: true, confidence: 0}
  3: Array(1)
   0: {transcript: "esto es una para sé",
       final: true, confidence: 0}
  4: Array(1)
   0: {transcript: "esto es una pasa",
       final: true, confidence: 0}
```

Fig. 6. Result in an incorrect and meaningless sentence.

Then we tried some incorrect sentences to know how the speech recognizer reacted to them. Following the previous sentence, we tried a completely unmeaningful variation of it. This was "esto es una pase", changing "frase" ("phrase") to "pase" (a verb). The sentence "esto es una pase" has no meaning in Spanish, although all its words are correct. In computing terms, it could pass the lexical and syntactical analysis of a language compiler, but not the semantic one. The results of the speech recognizer in this case are shown in Fig. 6.

In this case, the speech recognizer has considerably trouble in finding a match of what has been said. Instead of just one result, it returns five alternatives. It should be noted that the best result has also a lower confidence value than before. This example is interesting when illustrating how the recognizer works. Even when it has recognized the input, as result #2 shows, it discards it in favor of a meaningful sentence, despite the fact that it is not the correct answer.

In the above example we could pick the correct answer although it is not the one with higher confidence because we know what is 'wrongly' said in advance and we could check if we can find it inside the alternatives in the result. However, in a real situation we could not know in advance which could be the incorrectly pronounced words. Nevertheless, the array of results can give us some clues. If we discard the results that have more words than the sentence we expect to receive, we would still have 4 different results. If we compare them, we will see that all of them have the first three words in common "esto es una" and diverge on the last one. Hence, the recognizer is quite sure that the sentence said starts with "esto es una" but not so sure about the last one. Putting it in a form of array of coincidences in words among results it has the form of [4,4,4,0] in which each value represents the coincidence of the word in that position among results. Using this array, we can conclude that there is a problem at pronouncing the last word. Unfortunately, we cannot be sure of the precise issue, but at least we can detect that this word and not the others need more help and practice.



Ver si la decimos bien

Toca el micro y dila

ESTO ES UNA FRASE

Puedes probar de nuevo la frase o aquí debajo puedes practicar las palabras sueltas que parece que han salido mal

Fig. 7. Results obtained when some word is not correctly pronounced.

The behavior of the app in this case can be seen in Fig. 7. The app warns about a pronunciation problem with the sentence, highlighting the word that is not correctly identified. It also offers the possibility of practicing only the word that seems to be an issue.

## V. MarLuc in Use

MarLuc has been available on the Google Play Store for some years now. We wanted to share the tool freely with those who could find it useful as an complementary tool in speech therapy. No special marketing campaign has every been undertaken. In 2016, it included the possibility of checking exercises and pronunciation against the speech recognizer, but only with words. It was in 2018 when the app also included the possibility of checking of full sentences using speech recognizer. It has been downloaded more than 500 times. Over these years, the comments received from therapists and users have also helped to improve its design and exercises. Currently it is in use on 156 devices.

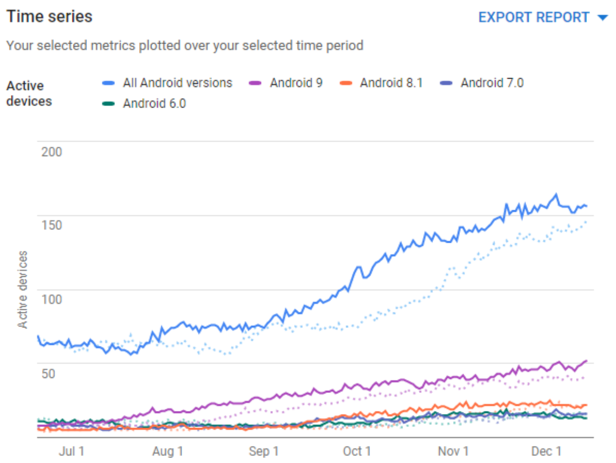


Fig. 8. Statistics of installed devices for MarLuc.

## VI. Conclusions

Native speech recognition capabilities of mobile and personal devices offer a free and widely available tool that could be used in speech therapy. Using public open source APIs and frameworks, we can go beyond the ordinary usage of speech recognition on our mobile devices and explore new ways to approach and improve speech therapy. There also some limitations that need to be considered, such as the difficulty of detecting the wrong phonemes and their positions in the words that are not being said correctly. This adds some complexity to detecting and making finely detailed diagnosis of the underlying problem in speech. Nevertheless, we can complement native speech recognition with some more knowledge in the code or by connecting the app to external tools to handle the knowledge management. Thus, we can make a more directed search of known issues with regards to speech production using, for example, specific sets of words for each case. This additional logic will also provide some context, which as shown, eases and improves the speech recognition process.

## Conflict of Interest

The authors declare no conflict of interest.

## Author Contributions

C. Alonso designed and implemented the app, conducted the data analysis and research, and co-wrote the paper. J.J. Astrain and T. Read co-supervised the work and co-write the paper.

## Acknowledgment

We wish to acknowledge the active and valuable contribution of therapists and families along the project.

## References

[1] National Institute on Deafness and Other Communication Disorders, Statistics on Voice, Speech, and Language. [Online]. Available: https://www.nidcd.nih.gov/health/statistics/statistics-voice-speech-and-language

[2] D. Crystal, *Patolog ía del Lenguaje*, C átedra, 1993.

[3] R. Paul, *Language Disorders from Infancy through Adolescence: Assessment Intervention*, St. Louis, MO: Mosby, Inc., 2001.

[4] M. Monfort, "Segundo debate en el IV Simposio de Logopedia. En M. Monfort, Ense ñar a hablar," in *Proc. CEPE*, 1995.

[5] A. Borrag án, J. Barrio, and J. Guti érrez, *El juego Vocal Para Prevenir*, Malaga: Aljibe, 1999.

[6] V. Acosta and A. Moreno, "Adquisici ón y desarrollo del lenguaje," *Enciclopedia de Educaci ón Infantil*, 2003.

[7] A. Kukulska-Hulme, *Mobile Language Learning Now and in the Future*, ISweden: Swedish Net University (N ätuniversitetet), 2006, pp. 295–310.

[8] A. Kukulska-Hulme, "Will mobile learning change language learning?" *ReCALL*, vol. 21, no. 2, pp. 157-165. 2009.

[9] S. S. Nash, "Mobile learning, cognitive architecture and the study of literature," *Issues in Informing Science and Information Technology*, vol. 4, pp. 811-818, 2007.

[10] J. Roschelle, "Unlocking the learning value of wireless mobile devices," *Journal of Computer Assisted Learning*, vol. 19, no. 3, pp. 260–272, 2003.

[11] A. Wasserman, "Software engineering issues for mobile application development," in *Proc. FSE/SDP Workshop on the Future of Software Engineering Research*, 2010.

[12] M. E. Joorabchi, A. Mesbah, and P. Kruchten, "Real challenges in mobile app development," in *Proc. ACM / IEEE International Symposium on Empirical Software Engineering and Measurement*, 2013, pp. 15-24.

[13] World Wide Web Consortium (W3C). HTML5 Specification. [Online]. Available: https://www.w3.org/TR/2011/WD-html5-20110405/

[14] Apache Cordoba. [Online]. Available: https://cordova.apache.org/

[15] Ionic Framework. [Online]. Available: https://ionicframework.com/

[16] S. Yusuf, *Ionic Framework by Example*, Packt Publishing Ltd, 2016.

[17] M. Huynh, P. Ghimire, and D. Truong, "Hybrid app approach: Could it mark the end of native app domination?" *Issues in Informing Science International Journal of Current Trends in Engineering & Research (IJCTER)*.

[18] K. A. More and M. P. Chandran, "Native vs hybrid apps," *International Journal of Current Trends in Engineering & Research (IJCTER)*, vol. 2, issue 6, pp. 563-572, June 2016.

[19] W3C Speech Api Community Group. [Online]. Available: https://www.w3.org/community/speech-api/

[20] Can I use? Web site for checking features on browsers. [Online]. Available: https://caniuse.com/#feat=speech-recognition

[21] S. MacDonald, Speech Recognition Plugin. [Online]. Available: https://github.com/macdonst/SpeechRecognitionPlugin

[22] Google, DialogFlow. [Online]. Available: https://dialogflow.com/

[23] Facebook, Wit.ai. [Online]. Available: https://wit.ai/

[24] Microsoft, LUIS Language Understanding, [Online]. Available: https://azure.microsoft.com/es-es/services/cognitive-services/language-understanding-intelligent-service/

[25] S. Janarthanam, *Hands-On Chatbots and Conversational UI Development: Build Chatbots and Voice User Interfaces with Chatfuel*, Dialogflow, Microsoft Bot Framework, Twilio, and Alexa Skills, Packt Publishing Ltd, 2017.

[26] A. Singh, K. Ramasubramanian, and S. Shivam, *Building an Enterprise Chatbot: Work with Protected Enterprise Data Using Open Source Frameworks*, Apress, 2019.

[27] M. Mitrevski, *Developing Conversational Interfaces for iOS*, Apress, 2018.

[28] Real Academia Espa ñola. Facts of the 23 ªEdition of Spanish Language Dictionary. [Online]. Available: https://www.rae.es/diccionario-de-la-lengua-espanola/presentacion

[29] DialogFlow Cordova/Javascript plugin. [Online]. Available: https://github.com/api-ai/api-ai-cordova

[30] Cordova-plugin-audioinput plugin. [Online]. Available: https://github.com/edimuj/cordova-plugin-audioinput.git

[31] Sphinx CMU website. [Online]. Available: https://cmusphinx.github.io/wiki/

**C. Alonso** is the head of Campus Services at Computing Services in the Public University of Navarre and a predoctoral researcher at the Mathematics, Statistics and Computing Science Department. He is hold with the MSc in computing science and MSc in electronics and control, and the BSc in computing science.

He had more than 25 years of experience in managing and developing applications. He is involved in activities directed to improve the life of persons with disabilities and towards universal accessibility. Orcid Id:

https://orcid.org/0000-0002-6178-4887

**Timothy Read** is a senior lecturer at UNED. He has held several positions in the university government. He is the cofounder of the ATLAS (Applying Technologies to LAnguageS) research group and has directed national and international funded projects on applying ICT to LSP and sub-languages. He is currently working in the area of MALL, language MOOCs, and their applications for social inclusion. He has also been a member of diverse scientific committees and has collaborated as an evaluator of national and international research project proposals.

**José Javier Astrain** received the bachelor's and master's degree in telecommunications engineering and the Ph.D. degree in computer science from the Public University of Navarre, Pamplona, Spain, in 1999 and 2004, respectively. He is currently a lecturer with the Public University of Navarre. His current research interests include distributed systems, and activities focused on improving life quality of persons with disabilities. Orcid Id: https://orcid.org/0000-0002-7792-6317.