Co-funded by the Horizon 2020
Framework Programme of the European Union

Deliverable D5.2
Virtual Assistant

Work Package 5: WP5

affecTive basEd iNtegrateD carE for betteR Quality of Life: TeNDER Project

Grant Agreement ID: 875325

Start date: 1 November 2019

End date: 31 October 2022

Funded under programme(s): H2020-SC1-DTH-2018-2020/H2020-SC1-DTH-2019

Topic: SC1-DTH-11-2019 Large Scale pilots of personalised & outcome based integrated care

Funding Scheme: IA - Innovation action

Disclaimer

The TeNDER consortium consists of the following Partners.

No	Name	Short name	Country
1	UNIVERSIDAD POLITECNICA DE MADRID	UPM	Spain
2	MAGGIOLI SPA	MAG	Italy
3	DATAWIZARD SRL	DW	Italy
4	UBIWHERE LDA	UBI	Portugal
5	ELGOLINE DOO	ELGO	Slovenia
6	ETHNIKO KENTRO EREVNAS KAI TECHNOLOGIKIS ANAPTYXIS	CERTH	Greece
7	VRIJE UNIVERSITEIT BRUSSEL	VUB	Belgium
8	FEDERATION EUROPEENNE DES HOPITAUX ET DES SOINS DE SANTE	HOPE	Belgium
9	SERVICIO MADRILENO DE SALUD	SERMAS	Spain
10	SCHON KLINIK BAD AIBLING SE & CO KG	SKBA	Germany
11	UNIVERSITA DEGLI STUDI DI ROMA TOR VERGATA	UNITOV	Italy
12	SLOVENSKO ZDRUZENJE ZA POMOC PRI DEMENCI - SPOMINCICA ALZHEIMER SLOVENIJA	SPO	Slovenia
13	ASOCIACION PARKINSON MADRID	APM	Spain

Document Information

Project short name and Grant Agreement ID	TeNDER (875325)
Work package	WP05
Deliverable number	D5.2
Deliverable title	Virtual Assistant
Responsible beneficiary	DW
Involved beneficiaries	UPM, UBI, CERTH
Type[1]	Report
Dissemination level[2]	PU
Contractual date of delivery	M20
Last update	01/10/2021

[1] **R:** Document, report; **DEM:** Demonstrator, pilot, prototype; **DEC:** Websites, patent filings , videos, etc.; **OTHER**; **ETHICS**: Ethics requirement; **ORDP**: Open Research Data Pilot.

[2] **PU:** Public; **CO:** Confidential, only for members of the consortium (including the Commission Services).

Document History

Version	Date	Status	Authors, Reviewers	Description
v 0.1	24/04/2020	Draft	Paride Criscio (DW)	Draft of initial Table of contents
v 0.2	20/09/2021	Draft	Paride Criscio, Camila Garcia (DW)	Input in section 1,2,3
v 1.0	21/09/2021	Draft	Luís Santos (UBI)	First Peer Review
V1.2	29/09/2021	Draft	Gregor Verdnik (ELGO)	Second Peer Review
V.1.3	01/10/2021	Draft	Camila Garcia (DW)	General overhaul of the document structure

Acronyms and Abbreviations

Abbreviation	Description
AI	Artificial Intelligence
API	Application Programming Interface
CLI	Command-line interface
DB	Database
EU	European Union
GA	Google Assistant
GDPR	General Data Protection Regulation
gTTS	Google Text-to-Speech
HAPI FHIR	HAPI Fast Healthcare Interoperability Resources
HLS	High Level Subsystems
ID	Identity
Mx	Month (where x defines a project month e.g. M8)
STFT	Short-time Fourier Transform
TCP	Transmission Control Protocol
TeNDER	affecTive basEd iNtegrateD carE for betteR Quality of Life
TTS	Text-to-speech Synthesis
TV	Television
Tx.x	Task
UI	User Interface
VPA	Virtual Personal Assistant
WPx	Work Package

Contents

List of Figures

1. INTRODUCTION

TeNDER Application aims to support patients with specific diseases and help them become more independent from other people in case of needing help or requesting services. The world's digitalization made sure that humans no longer need to contact anyone to seek help and are able to depend on a far more efficient and reliable device, which can take care of their everyday needs. Computers, mobiles, laptops, and other technological devices, which can carry out from simple to complex calculations and run programs to reduce monotonous work and waste of labor, became a part of our daily lives

Virtual Personal Assistant (VPA) has almost become a necessity in all electronic devices, to quickly provide solutions for the required problems. More than just being a bot, VPA can make life easier for the user in various ways. In general, tasks that virtual personal assistants might do are:

1. *Sorting through messages*: The average professional spends 28% of their day working on emails, and some of that time is no doubt spent deleting unnecessary correspondence. An example to take advantage of this is Gmail assistant, which provides a service to cluster emails into groups, including primary, social, and promotion. This simple change helps people stick to a philosophy called *Inbox Zero* by curating the emails that are more important and allowing them to process less vital categories, such as newsletters or social notifications, later.

2. *Managing your schedule*: The most difficult part of the hassle with scheduling meetings and appointments is finding a time that works for both parties. An example application for this task is Calendly [1], a popular app that allows people to book a time with someone based on when that person has free time. The user sets his/her availability, so they are not booked back-to-back.

   The user can integrate Calendly with Google, Outlook, Office 365, or iCloud calendars to avoid double bookings.

   Another application to take as a good example is Clockwise [2]. This app not only manages users' calendars, but it can also automatically move meetings to times that work better. The user can protect space for focused work, lunch, travel time or personal meetings.

   Clockwise resolves meeting conflicts, adjusts for time zones, and considers users' preferences, such as holding meetings during afternoons. This app may be better suited for teams that have to coordinate multiple schedules.

3. *Organize a trip*: Hopper [3] and TripIt [4] are two applications that assist users in managing their tasks: the first one can help saving money by predicting the best time to book a flight or hotel room and the second one creates a master itinerary, making information about dates and times, location, and confirmation codes easily accessible.

4. *Remind of important events and activities*: Even if some calendar apps have automatic reminders, users might need a heads-up in advance. There are plenty of options for apps here, for example, 24me [5]. It is a calendar, to-do list, and note app all in one, and it tells users what they need to do and when. For instance, it will

remind users to run an errand during their lunch hour, pay their bills, or pick up a card for a colleague's birthday next week. It also lets users complete some of the tasks from within the app, such as making a call, booking an airline ticket, or sending an email. Users can leverage the apps already integrated into their devices like Siri, Cortana, Alexa, or Google assistant, to manage reminders.

5. *Taking dictation*: Whether a user wants to make a quick note or draft a letter, they can use a dictation app to translate spoken words into text. There are several free and high-rated options to address this task, such as Otter [6], Speech [7], and Dictation Speechy Lite [8] is another real-time solution that uses AI for speech recognition for IOS-based devices. The last is Dictation.io [9], a web-based option that can help to dictate emails and documents using Google Chrome [10] web browser.
6. *Researching information*: Siri, Cortana, Alexa, or Google Assistant support quick web searches via voice command or trigger a specific functionality.

2. USER REQUIREMENTS

User requirements are based on the tasks that a virtual assistant can handle. TeNDER partners performed an analysis after the first pilot, during which they could already see some functionalities in action.
The review started with a collection of feedback from the users, considering where they had more issues navigating the app.

The Virtual Assistant module is meant to have different versions, from a basic idea of voice reproduction to the implementation of human interactions with the app and the mini pc. The aims of the module are not going to change through the versions. Instead, it will always try to facilitate the navigation for the TeNDER app targeted users (old people with specific diseases).

The analysis to implement a virtual assistant in TeNDER started with an overview of the most used tools, in which useful functionalities were identified, such as the ones mentioned below:

Amazon Alexa [11]: while not the first "Voice Assistant," Amazon Alexa was the first to drive this new wave of voice-first devices with the release of the Amazon Echo. Alexa's popularity quickly grew over a few short years with the help of multiple releases of new devices. This push of new devices continues to put Alexa on top, compared to the other assistants, as we see Alexa in more systems like phones, TVs, kitchen appliances, speakers, and more. With Alexa in more devices, the team at Amazon can train the AI with more data and continue to expand its skillset to empower developers with more capabilities to put in their custom skills and apps. There is now a huge community around developing Alexa Skills and pushing the limits of what the devices can do, and because of this, the entry barrier is very low.

Google Assistant [12]: although Alexa still holds a vast majority of the voice-first devices, Google Assistant usage and audience continues to grow by bringing its intelligent design to different fronts. It's built into the Android operating system of the most modern Android devices, making it easily accessible to Google's existing audience. On top of pushing its line of devices with the Google Home and Android devices, the Google Assistant's underlying toolset, Dialog Flow, is used by bots in many other systems such as Slack [13], Facebook Messenger [14], Skype [15], and more. This enables developers to harness the power of Google Assistant's understanding while expanding the reach of their custom skills and actions into other applications and services. Google also continues to train its AI in different ways to become more "human-like" which also forces the competition to do the same and to keep up with what is widely considered the smartest of the flagship AIs.

Samsung Bixby [16]: Bixby is Samsung's Virtual Assistant. Applications using Bixby are called Capsules and are available on any device where Bixby is present (Samsung Phones, select TVs and other appliances). Bixby was re-released in 2019 with Voicify [17], an elite partner working closely with the Samsung team to create the enterprise integration between the two systems. In 2021 it is believed that Samsung will deploy Bixby to nearly all the Billion+ devices globally that support the update.

Apple Siri [18]: Siri was one of the first flagship AIs released to the public as a part of iOS operating system. However, it was released with little ability to extend its functionality with custom skills. This limitation has caused great discontent amongst developers and brands as the power of voice assistants is to do things that wouldn't be considered, at times, possible with a native app. Having one UI limited by another is counterintuitive. So, it is not possible to create a custom, stand-alone voice app with Siri like it is with GA, Alexa & Cortana. Sirikit [19] was built to give iOS developers the ability to extend existing iOS apps, by leveraging the capabilities of the Siri AI.

Microsoft Cortana [20]: Microsoft closed their 3rd party application marketplace for Cortana. While the assistant is still active as part of their operating systems, 3rd party developers can no longer deploy to it.

From the functionalities of the aforementioned tool and the feedback given by users after the first TeNDER pilot execution, the design of the virtual assistant module has started.

3. VIRTUAL ASSISTANT FOR TeNDER

From TeNDER's project GRANT, the virtual assistant idea is described as task-focused on delivering a chit-chat virtual assistant, able to reply to common questions from both patients and caregivers, using the data coming from the analytics modules. In other words, this module uses the TeNDER interface to create a chit-chat interaction with the patient, intended to extract relevant information about the patient's emotional and physical condition. The interaction can be linked to the recommendation module to empower the patient about its health condition and create patient treatment adherence.
The virtual assistant is one of the four components of the HLS (High-Level Subsystems) and is responsible for extracting relevant information about patient status via reminders, questions, and matching services.
Concerning the TeNDER service toolbox, the Virtual Assistant belongs to Service cluster number five, which collects general information through TeNDER interfaces, specifically from patients and caregivers, to provide the following, exposed in the subsections below.

3.1 Overview of the virtual assistant in TeNDER app

In the vision of the TeNDER project, the virtual assistant is a software agent that performs tasks or services for an individual based on commands or questions. The interaction with TeNDER Virtual Assistant is performed via text and voice. The main idea is to interpret the speech using rules and AI and then provide the appropriate response. The response can be a voiced answer (yes/no, a number), or the display can directly show a particular service or information.

Speech recognition is one of the relatively new integrations into the VA. But, though it's moderately efficient, it is not very helpful and is not used by the users due to high error rates. Though the error percentage of the upcoming VPAs is around 5 percent, it still is not quite up to the mark where it becomes a part of the user's daily life. Thus, the project aims to build a VA with speech recognition that has aminimal error percentage. Under all those assumptions, the idea to insert a module to support the navigation is fully necessary and wanted by end-users.

One of the features unmissable in VA implementation is the voice recognition process. Voice recognition is a complex process using advanced concepts like neural networks and machine learning. The auditory input is processed, and a neural network with vectors for each letter and syllable is created, called a data set. When a person speaks, the device compares it to this vector, and the different syllables are pulled out according to the highest correspondence.

3.2 Technical features for the first version of the virtual assistant implementation

The first version of the TeNDER Virtual Assistant is implemented using three parts: the mobile application, the speaker, and the mini pc installed in the user's place.
The speaker will be able to play specific outputs to make the users know information without having the application opened or active.

This first version is developed in python and integrated with the TeNDER Ecosystem using the HAPI FHIR APIs to collect information to play via the speaker.

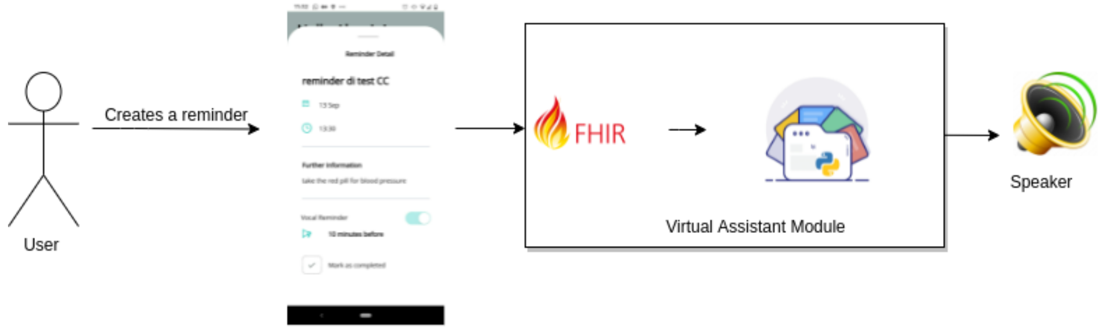

Figure 1 Simple diagram explaining the first version of the virtual assistant process

3.2.1 What the first version of VA can do

The first implementation of the virtual assistant will provide the announcements using a text-to-speech mechanism.

The speech played by the virtual assistant can be:

- The output of a reminder (e.g.: It's time to take the blue pill);
- The output of the recommender system module.

3.2.2 Functionalities and implementation related to VA 1st version

The module is designed to query HAPI FHIR's API to get the list of reminders or recommendations with the specific VA field set as true and with the patient's ID. The Vocal Reminder's option is defined from the interface directly by the user and specifies if the reminder has to be played via the speaker.

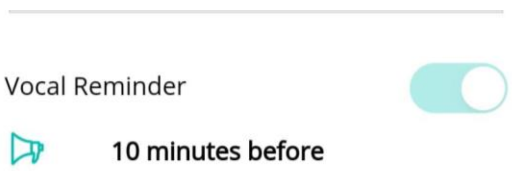

Figure 2 Virtual Assistant settings in the TeNDER mobile interface

The requested information from the HAPI FHIR is saved in a local Redis DB used as a cache to speed up the Virtual Assistant process and to avoid overenthusiastic polling of the main TeNDER DB.

Redis is most of the time referred to as a data structures server. This means that Redis provides access to mutable data structures via a set of commands, which are sent using a server-client model with TCP sockets and a simple protocol. Therefore, different processes can query and modify the same data structures in a shared way. Data structures implemented into Redis have a few special properties:

- Redis cares to store them on disk, even if they are always served and modified into the server memory. This means that Redis is fast, but it is also non-volatile;
- The implementation of data structures emphasizes memory efficiency, so data structures inside Redis will likely use less memory compared to the same data structure modeled using a high-level programming language.
- Redis offers several natural features for database administration, like replication, tunable levels of durability, clustering, and high availability.

Once the information is collected from the local DB, a specific text of a reminder or a recommendation, is played as a voice through the speaker, using gTTS (Google Text-to-Speech). gTTS is a Python library and CLI tool to interact with Google Translate's text-to-speech API. It writes speech data to a mp3 file, a file-like object (byte string) for further audio manipulation, or stdout.

3.2.2.1 Text-to-speech (TTS) technology

Usually, computers work in three distinct stages called input (often with a keyboard or mouse), processing (where the computer responds to the input), and output (where you get to see how the computer has processed your input, typically on a screen or printed out on paper).

Speech synthesis is a form of output where a computer or other machine reads words to you out loud in a real or simulated voice played through a loudspeaker; the technology is often called text-to-speech (TTS). It allows performing voice starting, a type of assistive technology that reads digital text aloud, from a machine-readable text. TTS can take words on a computer or other digital device and convert them into audio. TTS is very helpful for kids and adults who struggle with reading, and it can also help with writing and editing and even with focusing.

Speech synthesis works in the following way, given a paragraph of written text to play via voice. There are essentially three stages involved and three important factors in the whole process. The three factors or definitions to understand the process are:

- **Phoneme:** A phoneme is the smallest unit of sound that makes a word's pronunciation and meaning different from another word;
- **Prosody:** The patterns of rhythm and sound used in poetry;
- **Mel-spectrogram:** It is derived by applying a non-linear transformation to the frequency axis of short-time Fourier transform (STFT) of audio, to reduce the dimensionality. It emphasizes details in low frequencies, which are very important to distinguish speech, and de-emphasizes details in high frequencies, usually noisy.

Preprocessing must also tackle homographs, words pronounced in different ways according to what they mean. The word "*read*" can be pronounced either "*red*" or "*reed*," so a sentence such as "*I read the book*" is immediately problematic for a speech synthesizer. But if it can figure out that the preceding text is entirely in the past tense, by recognizing past-tense verbs ("I got up…, I took a shower…, I had breakfast…, I read a book..."), it can make a reasonable guess that "I read [*red*] a book" is probably correct. Likewise, if the preceding text is "I get up…, I take a shower…, I have breakfast..." the correct guess should be "I read [*reed*] a book."

This is a high-level diagram of different components used in the TTS system. The input to this model is text, which passes through several blocks and eventually is converted to audio.

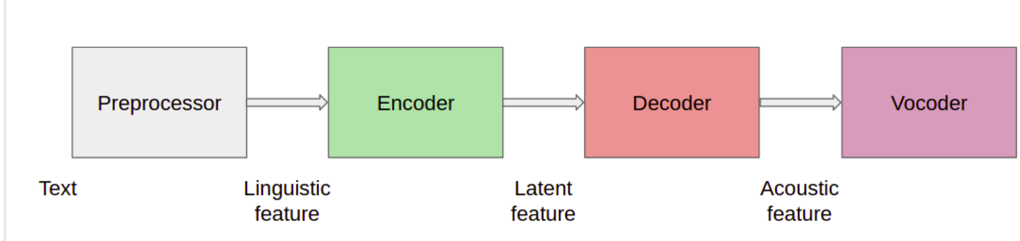

Figure 3 High-level diagram of components used in the TTS system

The first block in Figure 3 is the Preprocessor and is described with the following components:

- **Tokens**: Tokenize a sentence into words;
- **Phonemes/Pronunciation**: It breaks input text into phonemes based on their pronunciation. For example, "Hello, have a good day" converts to HH AH0 L OW1, HH AE1 V AH0 G UH1 D D EY1;
- **Phoneme duration**: Represents the total time taken by each phoneme in the audio;
- **Pitch**: Key feature to convey emotions, it affects the speech prosody;
- **Energy**: Indicates frame-level magnitude of Mel-spectrograms and directly affects the volume and prosody of speech.

The Linguistic feature only contains phonemes. Energy, pitch, and duration are the keys used to train the energy predictor, the pitch predictor, and the duration predictor, respectively. Additionally, they are used by the model to get a more natural output.
The second block, called Encoder, takes as input linguistic features (phonemes) and returns as output n-dimensional embedding. This embedding between the encoder and decoder is known as the latent feature. Latent features are crucial because other features like speaker embedding are concatenated with these and passed to the decoder. Furthermore, the latent features are also used for the prediction of energy, pitch, and duration, which in turn play a crucial role in controlling the naturalness of the audio.

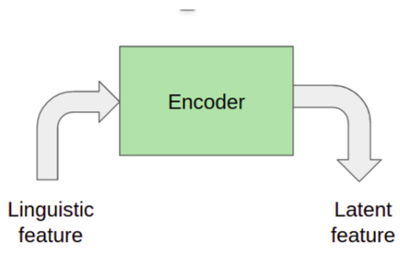

Figure 4 Input and output of the Encoder

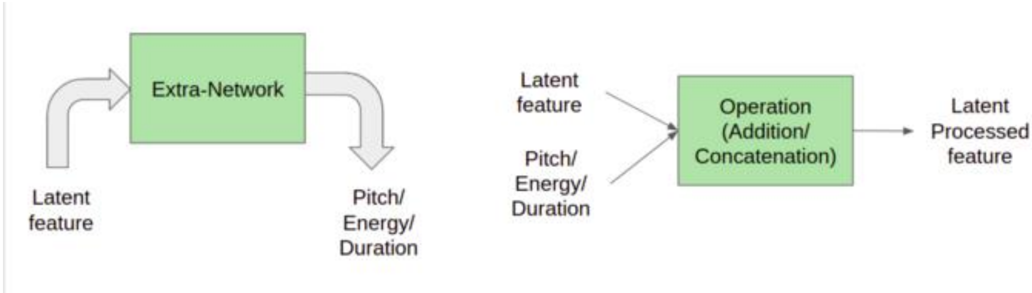

Figure 4 Components of the preprocessor block involved in the encoder block

The third block in the TTS diagram is called Decoder, and it is used to convert information embedded in the latent processed feature to the acoustic feature, i.e. Mel-spectrogram. The reason why output Mel-spectrograms are used instead of directly producing the speech/audio from the decoder is that audio contains more variance information (e.g., phase) than Mel-spectrograms. It causes an enormous information gap between the input and output for text-to-audio as compared to text-to-spectrogram generation.

The last block is named Vocoder and converts the acoustic feature (Mel-spectrogram) to waveform output (audio). It can be done using a mathematical model like Griffin Lim, a phase reconstruction method based on the redundancy of the short-time Fourier transform. Otherwise, this process can be done by training a neural network to learn the mapping from Mel-spectrogram to waveforms. Learning-based methods usually outperform the Griffin Lim method.

To directly predict waveform with the decoder and a learning-based method, the task must be split into two stages. The first is predicting Mel-spectrogram from latent processed features and then generating audio using Mel-spectrogram.

3.2.2.2. Reminder System and Recommender system module using VA

The reminder system in TeNDER allows users to have a calendar and create events to always have in mind what is important. The first version of the Virtual Assistant supports this functionality, specifying in the reminder details an option to play the reminder through the speaker.

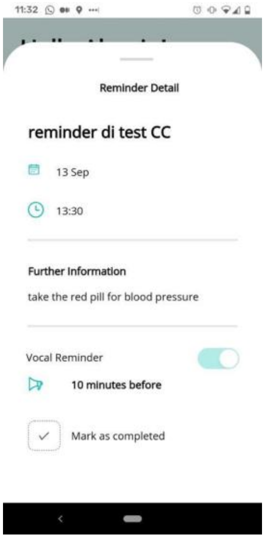

Figure 5 Reminder detail screen with vocal reminder activated

3.3 The second version of the Virtual Assistant implementation

The second version focuses on speech recognition technology that uses natural language processing to transform spoken language into a machine-readable format. The recognized text triggers a particular action, useful to the user.
The main aim of the VA module is to provide faster answers and improve the usability of the app, in particular for users that find issues using the TeNDER app.

The VA modules can support both patients and caregivers. The module means to be divided into two parts:

- **Proactive part:** aims to increase knowledge of the user's status. For example, the VA can ask the patient if they feel alright;
- **Reactive part:** focused on listening to a command, preprocessing it and providing an answer to the user. The voice command has to be transformed into a machine-readable text, and the output of a classification system will trigger the matching service. The TeNDER app can provide a precise answer (using speech synthesis) or show statistics or a particular section with the requested information.

The preconditions for VA module implementation are:

- A list of questions the system has to recognize, which can be useful to caregivers or patients;
- A list of services/answers matching the previous list of questions.

These lists can be different based on the patient's particular illness, some examples of questions are:

1. Question: You have a medical examination today at X. Would you like to engage in some activities before? Would you like me to inform someone?
2. Question: You have not taken your medication yet. Would you like to take it now?

3. Question: I detected that you did not sleep well last night. Would you like to take a nap after lunch? or Would you like to change your plans for today?
4. Question: You have used the toilet X times last night. Maybe you should drink more in the morning and less in the evening? Would you like this to be checked by a doctor?
5. Question: I detect that you are emotional. Would you like to listen to relaxing music? or Would you like to be accompanied by someone?
6. Question: Yesterday, you stayed in room X a lot . Would you like to go for a walk today? orYou are moving in the room in a way that is repeating. Would you like to call a caregiver?
7. Question: The entrance door is open. Would you like to close it?
8. Question: The light is out, but I detect that you are in the room. Would you like (me) to turn it on? or The humidity in your room has fallen. Are you comfortable? Would you like to call a caregiver?
9. Question: I detected that you have fallen. Can you get up? Would you like me to alert someone?
10. Question: I recognize you are wandering. Would you like to sit down and listen to music or watch TV? Would you like to talk to someone?
11. Question: Do you feel comfortable with the current temperature, or would you like me to increase it or decrease it?

3.4 Drawbacks and risks

There are many issues in the Virtual Assistant process, in particular, linked to the TTS technology.

The main constraint is that written text is ambiguous: the same written information can often mean more than one thing. So, the initial stage in speech synthesis, which is generally called pre-processing or normalization, is all about reducing ambiguity. Pre-processing involves going through the text and cleaning it up so the computer makes fewer mistakes when it reads the words aloud. Things like numbers, dates, times, abbreviations, acronyms, and special characters (such as currency symbols) need to be turned into words, which requires significant effort. The number 1843 might refer to several items ("one thousand eight hundred and forty-three"), a year or a time ("eighteen forty-three"), or a padlock combination ("one eight four three"), each of which is read out slightly differently. While humans follow the sense of what's written and figure out the pronunciation that way, computers generally don't have the power to do that. Therefore, they have to use statistical probability techniques (typically Hidden Markov Models) or neural networks (computer programs structured like arrays of brain cells that learn to recognize patterns) to arrive at the most likely pronunciation. In this sense, if the word "year" occurs in the same sentence as "1843," it might be reasonable to guess this is a date and pronounce it "eighteen forty-three." If there were a decimal point before the numbers (".843"), they would need to be read differently - as "eight four three."

In particular, the text-to-speech of gTTS is not precise for the Slovenian language. A possible solution was suggested by Slovenian partners called PONS [21], one of the leading language publishers from Germany. Since 1978, PONS has been publishing green dictionaries and language learning products for thirtytwo different languages.
The PONS online dictionary can be integrated ad-free into applications via the PONS API and provided with in-house terminology.

4. CONCLUSIONS

The first implementation of the virtual assistant comes from the exploration of state-of-the-art virtual assistants and the feedback provided by end-users after the TeNDER's First pilot execution. The Virtual Assistant is a continuously growing module, and this deliverable provides information about both Virtual Assistant versions that will be presented in TeNDER and a detailed description of the first implementation. The last version of this module will be updated and collected in deliverable 5.5, "Final version of the Final version of TeNDER Services".

REFERENCES

[1] https://calendly.com/

[2] https://www.getclockwise.com/

[3] https://www.hopper.com/

[4] https://www.tripit.com/web

[5] https://www.twentyfour.me/

[6] https://otter.ai/

[7] https://speechy.app

[8] https://apps.apple.com/us/app/voice-dictation-speechy-lite/id1239150966

[9] https://dictation.io/

[10] https://www.google.com/chrome/

[11] https://alexa.amazon.com/

[12] https://assistant.google.com/

[13] https://slack.com

[14] https://www.facebook.com/messenger/

[15] https://www.skype.com/

[16] https://www.samsung.com/us/explore/bixby/

[17] https://voicify.com/

[18] https://www.apple.com/siri/

[19] https://developer.apple.com/documentation/sirikit

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

[21] https://en.pons.com/translate