



Expertise
and insight
for the future

Tünde Taba

Personalized AI Assistant

Metropolia University of Applied Sciences

Bachelor of Engineering

Information and Communication Technology

Thesis

7.4.2019

Author Title	Tünde Taba Personalized AI Assistant
Number of Pages Date	21 pages 11 April 2019
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Professional Major	Mobile Solutions
Instructor	Merja Bauters, Lecturer
<p>The objective of this thesis was to study software agents that are also called smart speakers and then set up a personal assistant that is customizable.</p> <p>The study started from finding out how speech recognition works and then looking into voice assistants' history and roots. Once the speech is converted to digital data, the program recognizes the phonemes and based on probability matches them with words. The words get analysed in groups of three to know the context. The study continued with learning what is a smart assistant is and how it can assist a user in various service areas. When studying their history, it was found out that they have been around for decades helping humans in various ways. Not forgetting the negative side of technology, determining the problems with smart systems was also part of the research. For example, information security and privacy issues seem to be a common problem with smart speakers.</p> <p>Once the research was done, Mycroft AI – an open source full feature voice platform – was installed on a virtual machine. Then custom skills were added to the software using templates and Mycroft's own skill generator.</p> <p>The research on personal assistants and instructions on how to create a custom skill for one is a good starting point for someone new to the subject.</p>	
Keywords	AI, VDA, IVA, intelligent virtual assistant

Tekijä Otsikko	Tünde Taba Henkilökohtainen AI-avustaja
Sivumäärä Aika	21 sivua 11.4.2019
Tutkinto	Insinööri (AMK)
Tutkinto-ohjelma	Tieto- ja viestintäteknikka
Ammatillinen pääaine	Mobiilisovellukset
Ohjaaja	Lehtori Merja Bauters
<p>Insinööriyön tarkoituksena oli tutkia älykkäitä virtuaalisia avustajia, eli virtuaaliassistentteja ja älykaiuttimia, sekä luoda henkilökohtainen avustaja ja antaa insinööriyöraportin lukijalle helpot ohjeet sen tekemiseen.</p> <p>Aluksi tutkittiin, miten puheen ymmärtäminen toimii ja miltä näyttää puheentunnistusohjelmistojen menneisyys. Saatiin selville, että puhe muuttuu analogisesta datasta digitaaliseksi, minkä jälkeen ohjelma tunnistaa foneemit ja todennäköisyyden perusteella päättelee sanat. Sanat analysoidaan kolmen ryhmässä, jotta saadaan selville konteksti. Puheen tunnistamisen jälkeen perehdyttiin siihen, mitä ovat virtuaaliassistentit ja millainen on niiden historia. Tutkimuksen myötä selvisi, että erilaiset älykkäät virtuaaliset assistentit ovat olleet läsnä jo vuosikymmeniä auttamassa ihmisiä. Sitten otettiin selvää, miten määritellään se, mikä lasketaan tekoälyksi ja mihin assistenttien älykkyyttä voidaan käyttää. Selvisi, että assistentti pystyy melkein mihin vain, jos sitä osaa opettaa oikein. Jotkin tehtävät kuitenkin vaativat sen verran paljon yksityiskohtien ja logiikan työstämistä, että ne on parempi yksinkertaistaa tai jättää pois. Selvitettiin myös, mitä ongelmia ilmenee älykkäiden kaiuttimien kanssa esimerkiksi tietoturvaan ja yksityisyyteen liittyen.</p> <p>Kun oli luotu käsitys siitä, mikä on älykäs virtuaalinen assistentti, ryhdyttiin rakentamaan henkilökohtaista assistenttia. Pohjana käytettiin Mycroft AI -assistenttia, joka asennettiin virtuaaliseen tietokoneeseen. Assistentille opetettiin sama taito kolmella eri tavalla: ensiksi käytettiin tietokoneeseen asennettua ohjelmaa, toiseksi tehtiin Mycroftin omalla generointiskriptillä yksinkertainen versio ja kolmanneksi tehtiin monimutkaisempi taito, joka hakee tietoa internetistä.</p> <p>Tämä insinööriyöraportti on suunniteltu niille, jotka haluavat tietää lisää virtuaalisista assistenteista ja jotka haluaisivat helpon lähtökohdan henkilökohtaisen assistentin rakentamiselle. Esimerkit tehtiin yksinkertaisiksi, jotta kuka tahansa voi päästä mukaan matalalla kynnyksellä.</p>	
Avainsanat	AI, VDA, IVA, virtuaalinen assistentti, tekoäly

Contents

List of Abbreviations

1	Introduction	1
2	Virtual digital assistants	1
2.1	Speech recognition	2
2.2	History of voice assistants	4
2.3	Intelligence and responding to needs	5
2.4	Problems	8
2.5	The future of virtual assistants	9
3	Personalized virtual assistant	10
3.1	Mycroft – open source voice assistant	10
3.2	Setting up a virtual machine	11
3.3	Installing Mycroft	12
3.4	Creating a skill from Template	14
3.5	Creating a skill using Mycroft Skills Kit	17
3.6	Fortune cookie skill using an API	18
3.7	Conclusion	20
4	Summary	20
	References	22

List of Abbreviations

AI	Artificial Intelligence
VDA	Virtual digital assistant
IVA	Intelligent virtual assistant
UNIX	Hardware-independent operating system
MSK	Mycroft Skills Kit. Made for creating skills for Mycroft AI platform.
API	Application Programming Interface. Routines, protocols and tools for building software applications.
JSON	JavaScript Object Notation. A file-format with objects in arrays.

1 Introduction

When studying the history of virtual assistants, the past few years have shown a drastic change in the development and success of these nonhuman helpers. Nowadays most major technology companies have their own virtual assistant, or at least have started to develop one. The sci-fi fantasy of having a smart computer to talk to is no longer far in the future, but in the present. Modern AI assistants try to analyze and learn from the user's behavior so that they can grow to be more efficient and user-friendly as time goes by.

There are multiple popular smart assistants lined up to help us, but how can one tell which one is the most suitable for them? Finding the perfect AI assistant starts from deciding what the user wants to use the assistant for. After having a clear image about the required skills comes the research. Thus, what can one do if they cannot find the product they are looking for? The answer is to build one.

When a technology becomes viral and the competition is severe, the chances are that open source equivalents exist. These open software designs can be used as a baseline for a personalized intelligent personal assistant. Creating a personal AI assistant takes time and effort, but it has its perks, such as customization and data privacy.

The purpose of this thesis was to help the reader understand what virtual assistants do and how to create a personalized virtual assistant of their own. However, instead of studying only at the modern era, the thesis starts from the beginning of the voice assistants to really understand how long the journey has been to reach the current smart speaker revolution. This way the reader will grasp the concept of voice assistants, history and future included.

2 Virtual digital assistants

Virtual assistants are software applications that possess elements of modern artificial intelligence and have been created to interact with humans through voice. Large technology companies have their own intelligent virtual assistants for different purposes

and platforms. At first the main purpose of such an assistant was to answer the user's questions, but a point has been reached, where it is time to enhance it further with more capabilities and personalities. [1.]

2.1 Speech recognition

Speech recognition is the ability to recognize and react to spoken commands, which enables hands-free control, and is one of the core skills of a virtual assistant. Speech recognition starts from a microphone, which translates the vibrations into a wavelike electrical signal, and is converted by a sound card into a digital signal. The speech recognition software will analyze the digital signal and separate the phonemes that are needed to form words. Words can sound alike; thus, some natural language processing is required to figure the correct word. One way to select the correct one is to determine the context through a trigram analysis, which uses a database of frequent three-word clusters. With statistical analysis of texts, it is possible to count the frequency of monograms, bigrams, trigrams, and word-level trigrams. [2.]

Let's use the previous paragraph as an example of natural language processing and cryptographic frequency analysis. Breaking the paragraph down to frequencies shows that the characters e, t and a are the most common ones, as seen on Figure 1. The occurrence of pairs of characters can be seen in Figure 2 and lastly the trigram counts on Figure 3. Trigram analysis is first figuring out the possible characters in groups of three to form words, and then finding out the sentence in groups of three to derive the phrases. For a virtual assistant to be considered intelligent, simple speech recognition is not enough. Reacting to the phrases and learning as time goes by are what makes an assistant good. Modern powerful personal devices allow the assistants to recognize speech patterns and increase accuracy. [2.]

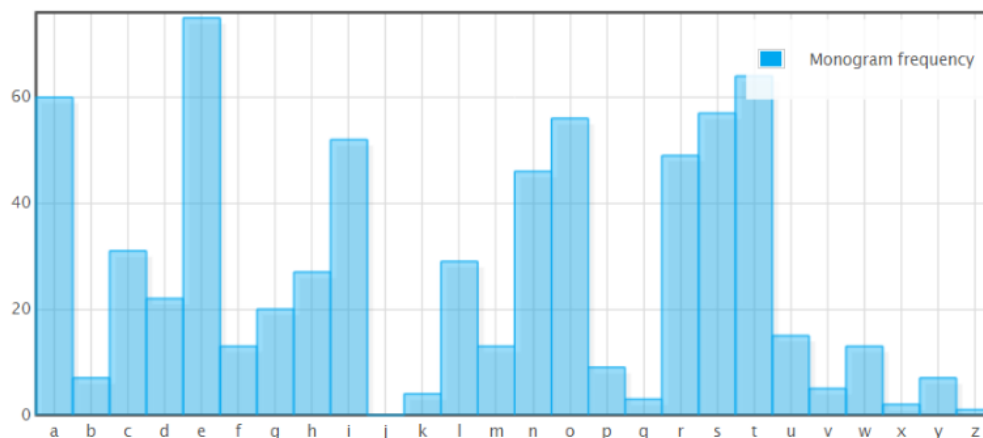


Figure 1. Monogram frequency in speech recognition chapter's first paragraph

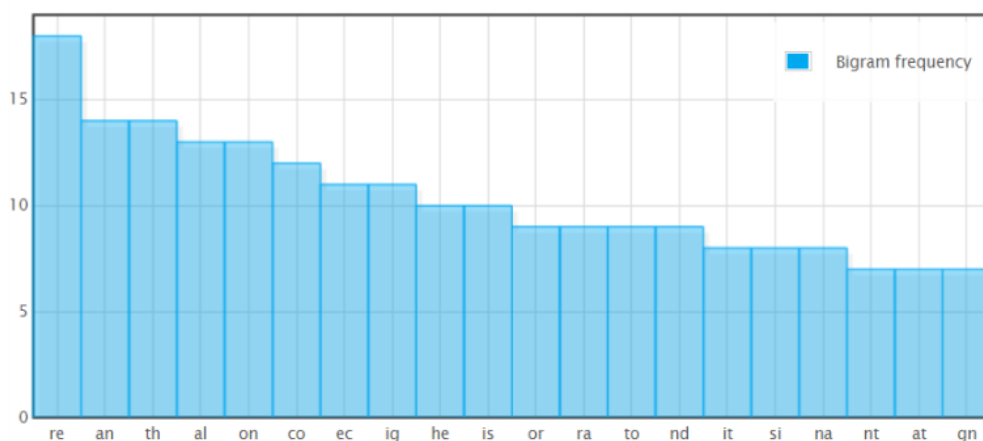


Figure 2. Bigram frequency in speech recognition chapter's first paragraph

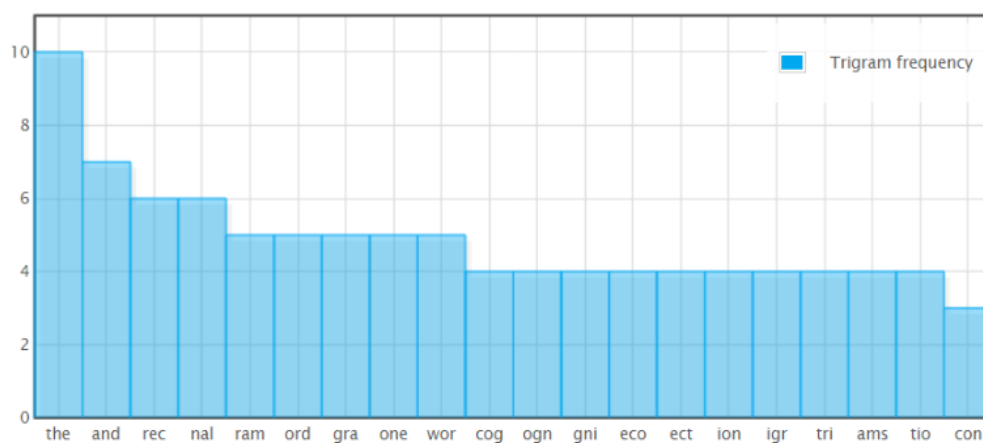


Figure 3. Trigram frequency in speech recognition chapter's first paragraph

2.2 History of voice assistants

The first attempt at recognizing voice commands was Radio Rex in 1922, a bulldog toy that could respond to its name by jumping out of its house. However, it had a low utterance accuracy, and rarely reacted to the required command. [3] The first digital speech recognition tool was the Audrey system by Bell Laboratories in 1953. Audrey was fully analogic and could recognize strings of digits. [4] In 1961 IBM introduced Shoebox, a machine with a vocabulary of 16 words and ten digits. Shoebox could solve arithmetic problems on voice command and print the answers. [5] The first speech recognition system with a 1000-word vocabulary was the Harpy system in 1976. It could understand various speakers with over 90% speech accuracy and was developed at Carnegie-Mellon University. [6] Dragon Dictate, the first speech recognition product designed for consumers, was launched in 1990 and cost \$9000. [7] In 1997 Microsoft introduced Windows Speech API on Windows 95 along with an office assistant that accepted speech input using Microsoft Speech Recognition Engine. [8] Apple introduced the well-known Siri assistant in 2011 for iOS 5. [9] In the same year IBM's Watson system won a quiz competition Jeopardy, where contestants were presented with answers to which they had to guess the question for. [10] From this point on the competition begins between the large tech companies and moving onto smart speakers' technology, as seen in Table 1.

Table 1. The smart speaker revolution

Voice Assistant/Smart Speaker Technology	Launched in
Google launches Google Now	2012 July
Microsoft introduces Cortana	2013 April
Amazon introduces Alexa & Amazon Echo	2014 November
Amazon launches Amazon Echo Dot & Amazon Tap	2016 March
Google launches Google Home	2016 November
LingLong launches DingDong	2016 November
Samsung introduces Bixby	2017 March
Cisco announces B2B Spark Assistant	2017 November
Apple launches HomePod	2018 February

Smart speakers are wireless speakerphones that can be controlled using voice commands. They possess artificial intelligence and can connect to the internet and various devices. These voice assistants are constantly waiting to hear the wake word, which triggers them to listen to a command. On a command the smart speaker can play music, set timers, control smart devices and even order pizza. Basically, if a speaker can do more than emit sound, it can be considered a smart speaker. [11.]

2.3 Intelligence and responding to needs

The purpose of a virtual assistant lies in its name; it is supposed to be an assistant that helps us in one way or another. The key to helping us in our daily lives is artificial intelligence, or more specifically machine learning to understand our preferences. Thus, what makes a computer intelligent? There is no one simple definition of AI, because there can be different kind of approaches to it. Intelligence can either refer to thinking and acting humanly or being rational and performing ideally. [12, 2] Most AI assistants are good at pretending to be intelligent, but is it considered artificial intelligence? For example, ELIZA – a program created by Joseph Weizenbaum in 1960s – simulated a conversation between a Rogerian psychoanalyst and a patient. It was supposed to create the illusion that the machine understands the conversation. Weizenbaum referred to it as a cocktail party conversation, where you do not understand what someone says, but because of the context, you are able to continue the conversation without getting caught. [13, 293]

At first glance ELIZA would seem to go under the acting humanly definition of intelligence, but it is missing some key capabilities according to Alan Turing's Turing test, which tests the machine's intelligent behavior. To pass the test ELIZA should store what it knows and use that information to answer questions, draw new conclusions and adapt to new circumstances. [12, 2-3] But even then, Hubert Dreyfus – the author of *What Computers Can't Do; A Critique of Artificial Reason* – would criticize the machine's artificial intelligence. Dreyfus proposed viewing the human behavior through subjective experience and phenomenology, which was brought forth by Heidegger, Merleau-Ponty, and others in the early 20th century. Dreyfus does not believe that a human mind can be replicated with equations, because the way it perceives the world is much more complicated. Dreyfus's phenomenological view stresses that a humans' perception

slides back and forth between the indeterminate outer horizon and less indeterminate inner horizon, while a machine must go from the details to the whole, because it is missing the inner horizon. [13, 212-215]

There are many different logics and formal languages. A formal language can be a programming language such as Python, while natural language is what humans use, such as English. Formal language can also mean logics, such as propositional logic or first-order logic. Propositional logic is a powerful but simple language for illustrating the basic concept of logic. It stores sentences about the world and infers new sentences that it uses to act. It recognizes propositions such as true, false and unknown. However, for an intelligent program to be able to represent complex environments and commonsense knowledge, it must use first-order logic. First-order logic is an expressive language to derive facts from other facts. While propositional logic only knows what exists in the world based on facts, first-order logic also takes into consideration objects and relations. [12, 285-290] In the beginning logical agents had a frame problem, which means that the information gets lost when not being able to tell what remains unchanged after an action. There are designs and solutions to prevent the frame problem, but it is still an issue that is present in AI. There is also another issue called qualification problem, which does not have a logical solution. System designers must specify what preconditions are needed for an action, what kind of exceptions there can be, and what details can be discarded. [12, 266-268]

1967 was the year of the knowledge-based programs, such as DENDRAL, MACSYMA, MacHack and LOGO. DENDRAL was the first so-called intelligent assistant that was put in use for its combined speed and knowledge equivalent to a chemistry Ph.D. [12, 318] The rest of the assistants were similar, but for different kind of purposes. MASCYMA was a mathematician's assistant, MacHack a chess-player program and LOGO an interactive learning environment. [13, 531]

One way of categorizing tasks a modern assistant can do is by placing them in five likely service areas: economy, personal, community, crowd and environment. Economy stands for transactions and financial management. Transactions include shopping on behalf of the user, recommending products and services, and reminding of upcoming needs. Financial management handles tax returns and household bills. It could also remind the user not to spend too much and alert about energy consumption. Personal area includes

health, behavior and daily living. An assistant can look after the user's emotional wellbeing, monitor body signs and synchronize them to medical information, help with diet and lifestyle changes, take notes and help understand moods and emotions. Daily living would mean reminding about household chores, synchronizing diaries and managing bureaucracies. Community area is about socializing and family. Social interactions could mean sharing time and place instructions prior to meeting up, alert that a friend is nearby, provide updates on what friends have been up to, which could work as a good conversation starter. Family is about bringing together schedules, synchronizing virtual assistants and security, such as knowing the children's whereabouts. Crowd area uses data gathered from the crowd, which include recommendations and connecting with people. Recommendations are based on integration of anonymized data from reviews and personal data, which may include warning about dangerous roads or telling if there is a famous pub nearby. Environment area can be divided to indoors and outdoors. Indoors means managing efficient running of utilities and appliances or manipulating the scent and music in the home to alter the user's mood. Outdoors means advice on gardening, alert about high pollen or air pollution and UV levels. [14, 2-22.]

An example of a task that a modern assistant could do for the user is finding a digital camera for under \$300 from anywhere on the internet. It may sound like an easy task at first, but there can be many complications. Nowadays websites include files and technology that makes it harder for a program to browse the pages. The program must understand where to look, how to follow relevant links and what to compare. However, for a program to be able to browse and think like a human, it would need to understand language fully and gradually update its knowledge. [12, 462-468.]

Artificial intelligence and virtual assistants can also aid people with disabilities, be caretakers and even satisfy social needs. Not having to move, touch or see anything when giving commands can be a huge benefit for those with disabilities. Smart speakers can also be companions that give encouragements and are there to listen. They can and have been programmed to react to depression and suicidal thoughts to help people with their inner struggles. [15] Amazon Alexa is starting to recognize moods, which changes the ways it reacts to commands. Recognizing moods allows more human-like conversations, but also dynamic targeted advertising. [16]

2.4 Problems

The intelligent assistants do not come without complications. To be a useful assistant, storing information about the user is crucial, which leads to privacy issues: who has access to the user's personal data and to the assistant in general? One issue that most manufacturers struggle with is controlling who has access to a smart speaker and its information. In the beginning of 2016 Google's Google Home supported several users, but they were all linked to the same Google account, which means that the personalized assistant made for the user no longer remained personalized. It learned from other user's as well and gave out private information. This was fixed and later Google Home could distinguish multiple users by voice and did not make all information available to everyone. [17] Amazon's Alexa has had its own complications as well; when there was a story about a girl ordering a mansion dollhouse using Alexa, some Alexa speakers that heard it on the news and did not have a password to make purchases also ordered a dollhouse. [18] Apple's HomePod that was released in 2018 did not learn from the competitors' earlier mistakes and does not recognize different voices. An assistant that cannot recognize its user is not very practical for personal requests. [19] However, even if the assistant can recognize voices, it is possible for someone to impersonate a user's voice and use it to make voice commands to unlock doors or order items. The unwanted voice commands are the easiest way to hack the assistant; for example anyone could set an annoying alarm if they wanted to. There has been case where even a parrot managed to order gift boxes through Alexa. Televisions can also trigger smart speakers intentionally or unintentionally. [20, 13-14]

Voice commands are not the only problem with IVA's. An IVA can be connected to the internet and other devices, which leads to security vulnerabilities. Not to mention the fact that the assistant may record private conversations and send them to the cloud without the user knowing. Another problem with internet is hackers. If the wi-fi is not secured properly, someone may attack the assistant and change the settings. [20, 15.]

There is also a bias problem with any kind of IVA that can learn things. The data fed to an assistant may not be filtered and neutral, and the level of intelligence these machines have does not recognize what is unfair, discriminating and unethical. The more advanced a system is the harder it is to predict what problems will occur in the future from the data

it has collected. Even if the creator has good intents, the AI system is only as good as the data it got from the user. [21.]

2.5 The future of virtual assistants

For some it takes time to get used to a software with a human-like voice to know everything about them, while others have already fallen in love with their virtual assistant. Either way, the assistants are growing at a rapid speed, both in technology and user-base, as seen in Figure 4. However, experts say that current smart speakers are not intelligent, because they cannot fully understand their users yet. Some of them may ask follow-up questions after a command and understand context quite well, but it is not enough if the commands get complicated. The race of which one has the smartest assistant has begun, which means that as time goes by the assistants will also get more intelligent. [22.]

Like smartphone apps, smart speakers have something similar called skills. For example, Amazon has a guide for developers to create skills for Alexa. The starting kit can convert speech to text, understand what the speaker means and even purchase products online. This way anyone can increase a smart speaker's value and expand its knowledge. Not only is this good for the product, but it also is a way for developers to make money. [23.]

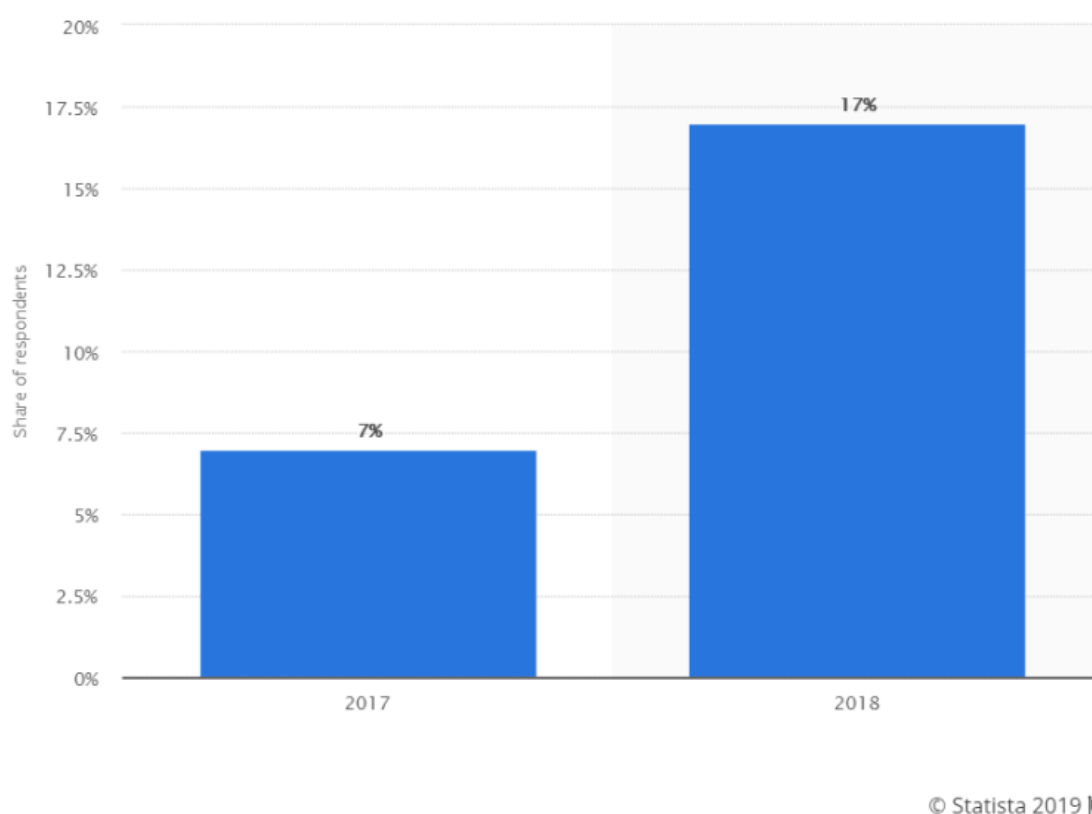


Figure 4. Smart speaker ownership rate among U.S. population in 2017 and 2018. [24.]

3 Personalized virtual assistant

3.1 Mycroft – open source voice assistant

To create a personal intelligent voice assistant, there is no need to start from scratch. There are some open-source projects that encourage collaboration and innovation. One of them is Mycroft's open platform, which will be used as an example. Mycroft is being advertised as being different from popular smart speakers, such as Alexa, because of its data privacy, customization, user agency and open data. The voice queries are not being saved and anyone can teach the assistant skills and join the project. [25.]

Mycroft is a good base for this project, because of its large user base. The documentation is also open-source, which means that it gets updated frequently and there are plenty of skills developed by other Mycroft users. The official forum is active and good for general discussion, feedback and support.

3.2 Setting up a virtual machine

There are various ways of setting up a Mycroft. Currently Raspberry Pi, Linux and Android are supported. For a Windows user, one way of developing on a Linux computer is using virtualization. Download and install Oracle VM VirtualBox and a freely distributed Linux ISO image suitable for the current processor architecture, for example Debian. VirtualBox is a virtualization program that can create and manage virtual machines with various operating systems. Launch VirtualBox and create a new virtual machine as seen in Figure 5.

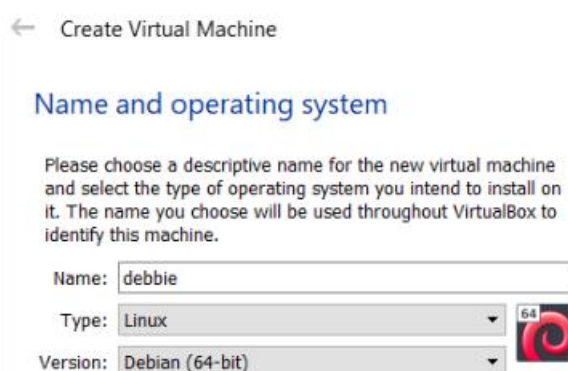


Figure 5. Creating a new virtual machine in VirtualBox

For this example, allocating 2048 MB memory is enough. From then on create a virtual hard disk, which is a VDI (VirtualBox Disk Image) and dynamically allocated. The size of the virtual hard disk can be 10 GB. After these steps the virtual machine is done. Select the freshly created machine and start it up. On start-up select a virtual optical disk file as seen in Figure 6. Insert the Debian ISO image that was downloaded before.

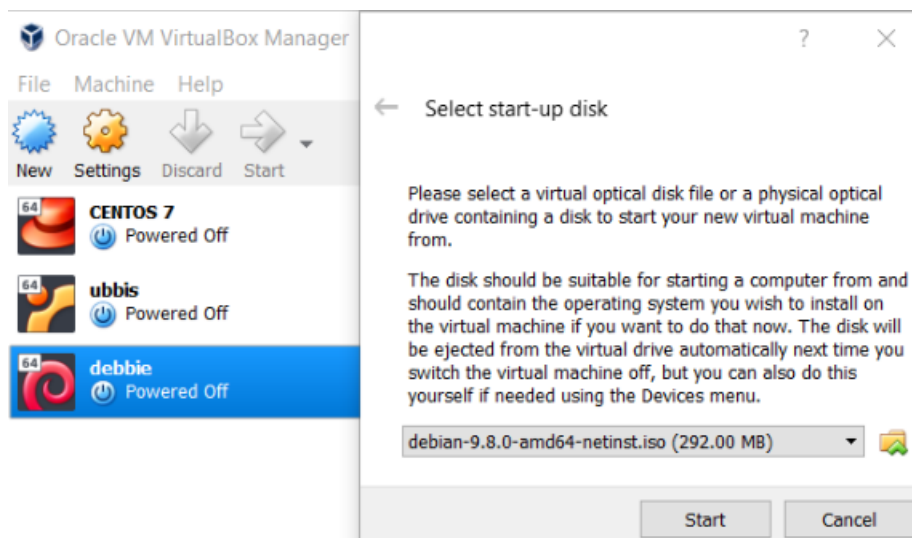


Figure 6. Adding ISO image to virtual machine

Next, for the installation of the operating system, select graphical install, preferred language, location and keyboard. The hostname can be anything and the domain name can be left empty. Next select a password for root and a user. After this it will pick the entire disk for partitioning with all the files. Make sure to select *yes* on writing changes to disk. The rest of the configuration can be run with the default settings. The installation will take a while. Once it has finished install the GRUB boot loader and select `/dev/sda`. Now Debian should be up and running.

3.3 Installing Mycroft

To be able to follow the examples properly basic knowledge of Unix environment and Git version control is needed. Knowing python programming language is also required to be able to write skills.

For a clean install of Debian, some packages must be installed first. Type the commands to install sudo package as seen in Listing 1. The sudo package allows users to have more privileges. Remember to use the root password set during the Debian installation.

```
youruser@debbie:~$ su -
Password:
root@debbie:~$ apt-get install sudo
```

Listing 1. Installing sudo package.

Since this is a virtual machine created for development purposes only it is fine to give all rights to the user so that it can write under /opt when installing Mycroft. Giving rights is easy using the command `visudo`, which will open a file to edit. Find the line for root privileges and add the user under it as seen in Listing 2.

```
# User privilege specification
root ALL=(ALL:ALL) ALL
youruser  ALL=(ALL:ALL) ALL
```

Listing 2. Giving rights to *youruser*.

Exit the editor (`Ctrl + X`), save (`Y`) and quit (`Enter`). To support running multiple GNOME terminals in one window, terminator is a good package to install and use instead of the default terminal, but it is not necessary for this example. Now with the required rights it is time to install git and clone the repository and set Mycroft up as seen in Listing 3. Change back to the user from root using the command `exit`. On Mycroft installation answer `Yes` to all options. The installation will take a few minutes.

```
youruser@debbie:~$ sudo apt-get install git
youruser@debbie:~$ git clone https://github.com/MycroftAI/mycroft-core.git
youruser@debbie:~$ cd mycroft-core
youruser@debbie:~/mycroft-core$ ./dev_setup.sh
```

Listing 3. Installing git, cloning Mycroft core repository and running setup script.

Now Mycroft is ready to be started and configured. Go to <https://home.mycroft.ai>, create an account and log in. Then go back to the terminal and use the command `./start-mycroft.sh debug`. It will boot up the assistant with a command line interface for debugging. The first run of Mycroft will take some time. Once Mycroft has loaded it will say that it is connected to the internet and ready to pair and give a code. If Mycroft never gives a code, try restarting the virtual machine and start it again. If Mycroft gave a code, go back to Mycroft Home and navigate to *You* and *Devices*. Add a device, insert the code and give the new device a name. Now Mycroft should be ready to be used and listed in devices as shown in Figure 7. Try asking *Hey Mycroft, what is the weather like?*

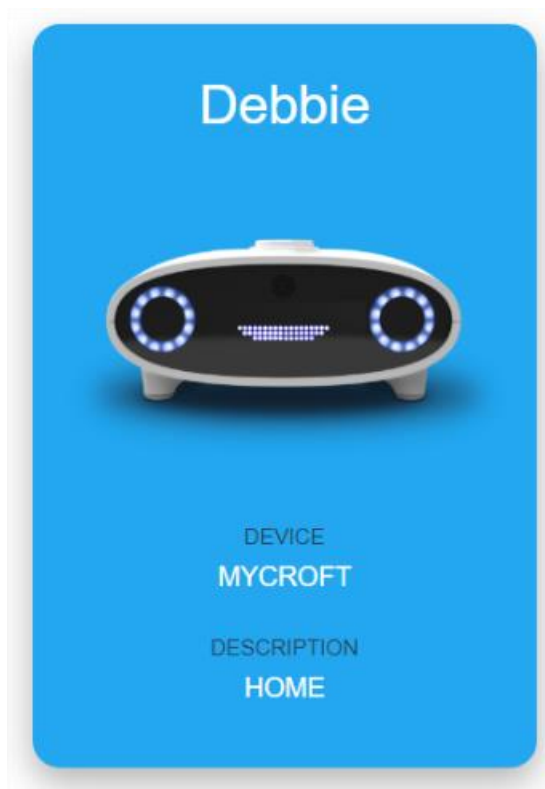


Figure 7. Mycroft device.

Under *General Settings* in the same menu as *Devices*, it is possible to select some preferences. Under *Advanced Settings* give a custom wake word to the assistant. Remember the phonemes mentioned earlier in speech recognition? They are needed when teaching the assistant its new name. For example, *Hello Debbie* becomes *HH AH L OW . D EH B IY .*

3.4 Creating a skill from Template

There are a variety of skills available either from Mycroft Marketplace or GitHub, but creating a custom skill is also possible. For a custom skill the requirement is to know some basic Python programming language. Creating a skill starts from an idea and planning on how the skill will work:

- What words or phrases will activate it?
- What will the assistant answer and do?
- What is needed for the task?

In this example, the answers are:

- User will ask for a fortune using the phrase *Tell me a fortune*.
- The assistant will give a fortune.
- The skill will need a Unix program called *fortune*, which shows a pseudorandom message.

Now that there is a skill planned, it is time to start developing it as seen in Listing 4. Install the fortune package and clone the mycroft-skills repository. Then copy the template to the skills directory and rename it to *skill-fortune*.

```
youruser@debbie:~/mycroft-core$ cd ~
youruser@debbie:~$ sudo apt-get install fortune
youruser@debbie:~$ git clone https://github.com/MycroftAI/mycroft-skills
youruser@debbie:~$ cd mycroft-skills
youruser@debbie:~/mycroft-skills$ cp -r 00__skill_template /opt/mycroft/skills
youruser@debbie:~$ cd /opt/mycroft/skills
youruser@debbie:~/opt/mycroft/skills$ mv 00__skill_template skill-fortune
youruser@debbie:~/opt/mycroft/skills$ cd skill-fortune
```

Listing 4. Installing fortune package, cloning Mycroft skills repository, and copying the template.

Some changes need to be done in the current directory. First create an intent, which is utterance that Mycroft will match with the new skill. For this, any kind of editor can be used, just make sure the path and filename match with Listing 5.

```
youruser@debbie:~/opt/mycroft/skills/skill-fortune$ nano vocab/en-us/Fortune.voc
```

Listing 5. Creating Fortune.voc file under vocab/en-us directory.

Proceed with adding the following lines. These words are what Mycroft will react to and start the skill:

- Fortune
- Tell me a fortune
- Cookie

Exit (Ctrl + X), save (Y) and quit (Enter) file. Now it is time to create a dialog. Dialog is a phrase that Mycroft will say. Ensure that the directory and filename are as in Listing 6.

```
youruser@debbie:~/opt/mycroft/skills/skill-fortune$ nano dialog/en-us/fortune.dialog
```

Listing 6. Creating `fortune.dialog` file under `dialog/en-us` directory.

Curly brackets indicate that the word is a variable. The variable will refer to the output the fortune program gives. Insert the following line to the file:

- Here is your fortune. {fortune}

Exit and save using the same commands as before. Then edit the `__init__.py` file in the fortune skill directory and copy Listing 7. Remember to stay consistent with indentation and use only either tabs or spaces to avoid errors. The copied code imports the required Mycroft core modules and subprocess, which can start an outside program. In this case it is the program called *fortune*.

```
import subprocess

from adapt.intent import IntentBuilder
from mycroft.skills.core import MycroftSkill, intent_handler
```

Listing 7. Importing the required modules.

Then copy the class from Listing 8. The intent handler requires the fortune intent. When hearing the intent, the program will run the subprocess *fortune*, and writes out the piped process, which is converted to string using a new line. The fortune dialog is given to the speak dialog function along with the standard output as the variable `fortune`. The stop function allows the user to terminate the skill by saying *stop*.

```
class Fortune(MycroftSkill):
    def __init__(self):
        super(Fortune, self).__init__(name='Fortune')

        @intent_handler(IntentBuilder('').require('Fortune'))
        def handle_fortune(self, message):
            ret = subprocess.run(['fortune'], stdout=subprocess.PIPE, universal_newlines=True)
            self.speak_dialog('fortune', data={'fortune': ret.stdout})

        def stop(self):
            pass

    def create_skill():
        return Fortune()
```

Listing 8. Fortune class

Now the skill should function; thus, start up Mycroft again with debugging command line interface as shown before. Once it is up and running, try talking to it using the wake word and saying the intents, or write them in the command line, as seen in Figure 8.

```
History =====
year. Take an elephant to lunch.
cookie
>> Here is your fortune. You will never know hunger.
fortune
>> Here is your fortune. Seeing that death, a
    necessary end, Will come when it will come. --
    William Shakespeare, "Julius Caesar"
```

Figure 8. Asking Mycroft for a fortune.

3.5 Creating a skill using Mycroft Skills Kit

Creating a skill using msk (Mycroft Skills Kit) is a quick way to get started. Run the script in mycroft core's bin directory, as seen in Listing 9, and it will prompt to answer a few questions regarding the skill.

```
youruser@debbie:~/mycroft-core$ cd bin
youruser@debbie:~/mycroft-core/bin$ ./mycroft-msk create
```

Listing 9. Running the msk create script.

First give the skill a short name followed by adding words and phrases that Mycroft uses to trigger the skill. Press Enter between the phrases and press Enter twice to move to the next question, which is what Mycroft should say when triggered. The last step is to give short and long descriptions to the skill. If there is a configured git user, it will also create the initial commit.

Now the skill has been initialized and should be in the skills folder with the rest of the skills. Run Mycroft with debugging command line interface and check out the results. If Mycroft is already running, reboot it to load the new skill.

3.6 Fortune cookie skill using an API

Let's take the fortune skill to the next level and use an API. <http://www.yerkee.com/api> has a fortune cookie API, which returns a cookie wrapped in JSON. It is also possible to retrieve a cookie from a specific category.

The first steps are the same as in the previous fortune skill. Start with a skill template and give an intent. This time the phrases will be more complicated. The round brackets mean that any of the words inside them are allowed in that space. The category will be something that the user completes; thus, it is marked as a variable. Change the previous fortune skills' intent so that they do not get confused with the new one. Add the following line to `TellMeFortune.intent`:

- Give me (a|an) {category} fortune
- Tell me (a|an) {category} fortune

Then add the dialogs the assistant will answer. This time there will be two: one for listing the categories in case the user messed up the intent and the other for giving the fortune. Add the following line to `NotFound.dialog`:

- Sorry, I couldn't find the category. Try one of the following: all, computers, cookie, definitions, miscellaneous, people, platitudes, politics, science and wisdom.

And the following to `HereIsFortune.dialog`:

- Your {category} fortune is {fortune}

Then open the `__init__.py` file and start with the imports and the request function. The `requests` package is needed to be able to send a request to the API and create a function to retrieve the fortune based on category. Create a function to retrieve a fortune based on category as seen in Listing 10.

```
import requests

from mycroft import MycroftSkill, intent_file_handler, intent_handler, \
    AdaptIntent
from mycroft.util.log import LOG
```

```

API_URL = 'http://www.yerkee.com/api/fortune/'

def get_fortune(category):
    r = requests.get(API_URL + category)
    if (200 <= r.status_code < 300):
        return r.json()['fortune']
    else:
        return None

```

Listing 10. Required imports and making a request to fortune cookie API.

Then add the skill class, which handles the intents and dialogs. It will also use the `get_fortune` function and insert the required category that the user said, as seen in Listing 11.

```

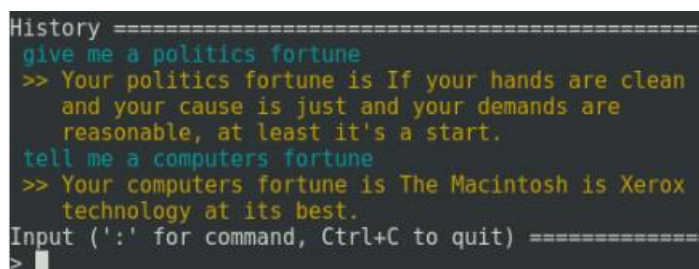
class FortuneSkill(MycroftSkill):
    @intent_file_handler('TellMeFortune.intent')
    def handle_fortune(self, message):
        category = message.data['category']
        fortune = get_fortune(category)
        if fortune:
            self.speak_dialog('HereIsFortune', {
                'category': category,
                'fortune': fortune})
        else:
            self.speak_dialog('NotFound')

def create_skill():
    return FortuneSkill()

```

Listing 11. Class that returns a fortune based on a category.

The new skill is ready to be used. Run Mycroft in debug mode and test it, as seen in Figure 9.



```

History =====
give me a politics fortune
>> Your politics fortune is If your hands are clean
and your cause is just and your demands are
reasonable, at least it's a start.
tell me a computers fortune
>> Your computers fortune is The Macintosh is Xerox
technology at its best.
Input (':' for command, Ctrl+C to quit) =====
>

```

Figure 9. Mycroft giving a fortune based on a category using an API.

3.7 Conclusion

Mycroft AI is fast to setup and simple to program with some basic understanding of UNIX environment and object-oriented programming. It is even easier to setup if the user does not want to customize it. The way skills work is easy to learn and there are not restrictions to what the skill can do. Starting from some very simple skills and working the way up to more complicated ones is a good method to determine what can be done, than proceeding straight into a complex algorithm. The speech recognition and response are fast with Mimic text to speech engine.

How intelligent the assistant turns out is entirely depending on the developer's skills, but the platform could be more flexible with its intents and keywords. Being forced to create an account to Home Mycroft and pairing it is an unpleasant step in the process. Regarding speech recognition, sometimes Mycroft has trouble understanding words and the user needs to speak slower and clearer than usual. Another problem is Mycroft reacting to its wake word way too easily and even a simple background noise may trigger it.

All things considered; Mycroft AI is a fun platform to play around with. It allows the developer to create an assistant with the voice, personality and skills that the user prefers. It is a good project to work on as a hobby, especially if it is taught skills that make it learn something from the user and get smarter as time goes by.

4 Summary

Speech recognition and assistants have been around for a long time, but they have not shown much growth regarding intelligence yet. Natural language processing and reacting to phrases are not new technologies themselves, but the technology around them has grown rapidly, which has allowed assistants to expand their tasks. Fifty years ago, a voice command could only make the program solve arithmetic problems, while in the modern world one voice assistant can perform thousands of tasks. Only now that smart speakers are getting popular has voice and mood recognitions become relevant.

What was learnt about artificial intelligence is that its definition can vary and be misleading in some cases. Some may refer to AI when talking about simple tasks like

decision-making or speech recognition that humans can do, while others refer to a machine simulating human intelligence. The definition is open because there is not one way to describe intelligence.

It is a fact that the more data a virtual assistant gathers the smarter it is, but the data may end up in wrong hands. If the user is not careful enough anyone can access their personal information or even order products without them knowing. A smart speaker may also accidentally react to something thinking it is a wake word and record the conversation. Connection to the internet always adds risks and gives room to hackers. It is important to make sure that the wi-fi is properly secured.

At first glance personal assistants may seem intelligent with their various skills. However, once the base is done, the actual skills are rather easy to program. Customizing a personal assistant may seem complicated, but with certain tools and basic programming knowledge it can be done in under a day.

References

- 1 Techopedia dictionary [Internet]. Techopedia™; Intelligent Virtual Assistant; [cited 2019 April 7]. Available from: <https://www.techopedia.com/definition/31383/intelligent-virtual-assistant>
- 2 Zwass V. Speech Recognition [Internet]. Encyclopædia Britannica Online: 2016; [cited 2019 April 7]. Available from: <https://www.britannica.com/technology/speech-recognition>
- 3 Markowitz J. Toys That Have a Voice. Speech Technology Magazine [Internet]. 2003 March [cited 2019 April 7]. Available from: <http://www.speechtechmag.com/Articles/PrintArticle.aspx?ArticleID=30031>
- 4 'Audrey' – New Electronic Device Developed by Bell Labs. Signal. 1953 Jan – Feb; 52 [cited 2019 April 7]. Available from: <https://play.google.com/books/reader?id=UEbAQAA-MAAJ&hl=en&pg=GBS.PA52>
- 5 IBM. IBM Shoebox. IBM Archives [Internet]. [cited 2019 April 7]. Available from: https://www.ibm.com/ibm/history/exhibits/specialprod1/specialprod1_7.html
- 6 Bisiani R. The Locust System. Simon J.C. (eds) Spoken Language Generation and Understanding. NATO Advanced Study Institutes Series (Series C — Mathematical and Physical Sciences), vol 59 June 26 – July 7, 1979. Springer Dordrecht; 1979.
- 7 Pinola M. Speech Recognition Through the Decades: How We Ended Up With Siri. PCWorld [Internet]. 2011 November 2 [Cited 2019 April 7]. Available from: https://www.pcworld.com/article/243060/speech_recognition_through_the_decades_how_we_ended_up_with_siri.html
- 8 Hunt A. SAPI: Microsoft Windows Speech API. Comp.Speech [Internet] 1997 April 21 [Cited 2019 April 7]. Available from: <http://www.speech.cs.cmu.edu/comp.speech/Section1/Interfaces/ms.speech.api.html>
- 9 Apple Launches iPhone 4S, iOS 5 & iCloud. Apple Press release [Internet]. 2011 October 4 [Cited 2019 April 7]. Available from: <https://www.apple.com/newsroom/2011/10/04Apple-Launches-iPhone-4S-iOS-5-iCloud/>
- 10 Dave Ferrucci at Computer History Museum: How it all began and what's next. IBM Research Blog [Internet]. 2011 December 1 [Cited 2019 April 7]. Available from: <https://www.ibm.com/blogs/research/2011/12/dave-ferrucci-at-computer-history-museum-how-it-all-began-and-whats-next/>
- 11 Rahman K. Smart speakers - everything you need to know. What Hi-Fi [Internet]. 2019 February 5 [Cited 2019 April 7]. Available from: <https://www.whathifi.com/advice/smart-speakers-everything-you-need-to-know>

- 12 Russel S, Norvig P. Artificial Intelligence: A Modern Approach (3rd Edition). Pearson; 2009 December 11.
- 13 McCorduck P. Machines Who Think (2nd Edition). A K Peters/CRC Press; 2004 March 19.
- 14 Institute of Management Studies, Mindshare. FUTURE_MEDIALAB: Assistant (VA) [Internet]. 2014 Sep 15 [Cited 2019 April 7]. Available from: [https://www.mindshareworld.com/sites/default/files/What Can I Help You With Virtual Assistant Report MindshareUK.pdf](https://www.mindshareworld.com/sites/default/files/What%20Can%20I%20Help%20You%20With%20Virtual%20Assistant%20Report%20MindshareUK.pdf)
- 15 Donnelly L. Digital assistants could alleviate the loneliness of elderly. Telegraph [Internet]. 2018 Jan 6 [Cited 2019 April 7]. Available from: <https://www.telegraph.co.uk/news/2018/01/06/digital-assistants-could-alleviate-loneliness-elderly/>
- 16 Fussell S. Alexa Wants to Know How You're Feeling Today. The Atlantic [Internet]. 2018 Oct 12 [Cited 2019 April 7]. Available from: <https://www.theatlantic.com/technology/archive/2018/10/alexa-emotion-detection-ai-surveillance/572884/>
- 17 Raphael JR. Google Home has a glaring flaw. Computerworld [Internet]. 2016 Nov 1 [Cited 2019 April 7]. Available from: <https://www.computerworld.com/article/3137187/google-home-glaring-flaw.html>
- 18 Liptak A. Amazon's Alexa started ordering people dollhouses after hearing its name on TV. The Verge [Internet]. 2017 Jan 7 [Cited 2019 April 7]. Available from: <https://www.theverge.com/2017/1/7/14200210/amazon-alexa-tech-news-anchor-order-dollhouse>
- 19 Patel N. Apple HomePod Review: Locked In. The Verge [Internet]. 2018 Feb 6 [Cited 2019 April 7]. Available from: <https://www.theverge.com/2018/2/6/16976906/apple-homepod-review-smart-speaker>
- 20 Wueest C. A guide to the security of voice-activated smart speakers (An ISTR Special Report). Symantec [Internet]. 2017 Nov [Cited 2019 April 7]. Available from: <https://www.symantec.com/content/dam/symantec/docs/security-center/white-papers/istr-security-voice-activated-smart-speakers-en.pdf>
- 21 Vanian J. Unmasking A.I.'s Bias Problem. Fortune [Internet]. 2018 Jun 25 [Cited 2019 April 11]. Available from: URL: <http://fortune.com/longform/ai-bias-problem/>
- 22 Charlton A. Hello, computer: As Alexa soars and Siri flags, what's next for the virtual assistant? GearBrain [Internet]. 2017 Nov 20 [Cited 2019 Apr 7]. Available from: <https://www.gearbrain.com/alex-siri-ai-virtual-assistant-2510997337.html>
- 23 Alexa Skills Kit. Amazon Alexa [Internet]. [Cited 2019 Apr 7] Available from: <https://developer.amazon.com/alexa-skills-kit>
- 24 Smart speaker ownership rate among U.S. population in 2017 and 2018. Statista [Internet]. 2018 July [Cited 2019 April 7]. Available from: <https://www-statista.com/statistics/797530/us-smart-speaker-ownership-rate-by-product/>

- 25 Mycroft Mark II Media Kit. Mycroft AI [Internet]. [Cited date 2019 Apr 7]. Available from: <https://mycroft.ai/media/>