Toward Development of A.D.A. – Advanced Distributed Assistant *

Fernando Freire[1], Thatiane Rosa[1,4], Guilherme Feulo [1], Carlos Elmadjian[1], Renato Cordeiro [1], Shayenne Moura[1], Acácio Andrade[1], Lucy Anne de Omena[1], Augusto Vicente[3], Felipe Marques[2], Aléxia Sheffer[2], Otávio Hideki[2], Patrícia Nascimento[1], Daniel Cordeiro[2], Alfredo Goldman[1]

[1] Institute of Mathematics and Statistics – IME/USP

[2]Schoool of Arts, Sciences and Humanities – EACH/USP

[3]Faculty of Philosophy, Linguistics, and Human Sciences – FFLCH/USP

[4]Federal Institute of Tocantins – IFTO

{acaciotda, lucy.omena, augusto.vicente}@usp.br

{felipercmarques, otaviohiga, ac.scheffer, daniel.cordeiro}@usp.br

{fernando.scattone, shayenne.moura, renatocf}@alumni.usp.br

{elmad, feulo, thatiane, gold}@ime.usp.br, pathilink@gmail.com

Abstract. *The A.D.A. – Advanced Distributed Assistant – project aims to build a smart distributed personal assistant, that is, a virtual agent that can interact with the user through an ecosystem of devices, such as IoT (Internet of Things), by voice commands in Portuguese. The project is divided into six scientific initiation subprojects from different areas of computer science, where each one is co-advised by a graduate student. An open source proof of concept is being created in order to demonstrate the assistant capabilities and its applications in public and private domains.*

1. Introduction

With the advance of machine learning techniques in the past decade and, in particular, with the success of end-to-end learning architectures, voice assistants ceased to be a concept to become a product. However, despite their growing commercial success, there are still important limitations to be addressed.

We can define a voice assistant as an artificial agent capable of recognizing verbal commands, extracting its meaning, and executing one or more tasks intended by the user [Mitchell et al. 1994]. In this way, the assistant plays, in practice, the role of simplifying the intermediate layer between the user's mental model and task accomplishment, releasing her or him from interacting with numerous different interfaces and frequent

*This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001, by Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), by Fundação de Amparo a Pesquisa de São Paulo (FAPESP) and by extension group CodeLab (uclab.xyz/site). Website: (uclab.xyz/ada). The name is also a tribute to Ada Augusta King, Countess of Lovelace, recognized as the first computer programmer in history, due to her work on the Charles Babbage's Analytical Machine [Lovelace and Toole 1992].

context changes involved in the process. This role can also be viewed in terms of Mark Weiser's "transparency" in ubiquitous computing [Weiser and Mark 1993]. That is because the assistant acts as a bridge to perform the requested task, hiding from the user all computational mechanisms that make it possible.

On the other hand, this convenience has a non-negligible infrastructure cost. That is because the agent must manage and adequately interact with the devices that integrate the user's ecosystem, which demands, among other aspects, interoperability of Application Programming Interfaces (API) and process scalability. All processing that must take place to turn voice commands into operations performed by devices needs to be modularized and distributed to deal with the growing users' demand for such systems.

Furthermore, almost all assistants available in the market are proprietary, which has raised several discussions regarding user privacy and security, in particular when it comes to the use of personal data for commercial purposes [Tankard 2016]. The few open-source alternatives, despite having more privacy safeguards, present a substantial worse user experience than their proprietary counterparts.

Another negative point with current assistants is what we call "platform restrictions". Due to their proprietary nature, most of these agents are enclosed in a single device, instead of having a distributed interface. These restrictions make it harder to create a universal API for sensors and actuators, as manufacturers are required to follow standards imposed by companies that control commercial assistants for device compatibility.

Finally, the virtual assistants' interaction problem is a field of research that remains largely unexplored. In addition to the lack of adequate support for Brazilian Portuguese — which is particularly important for our reality — it is not yet clear which techniques are suitable for resolving linguistic ambiguities in users' commands and the best mechanisms for minimizing and recovering from errors.

To deal with these interdisciplinary challenges, the objective of this project is to develop A.D.A.: "Advanced Distributed Assistant", an intelligent distributed personal assistant, based on free software and hardware, that must understand and process voice commands in Portuguese, and allow the user to interact with different smart devices.

The paper is organized as follows. Section 2 presents literature background on scientific fields used to develop an virtual assistant. Section 3 presents the A.D.A project goals, architecture and usage workflow. Section 4 describes challenges associated with development of virtual assistants. A discussion and comparision between most well known virtual assistants is done in Section 5. Section 6 discuss methodology used in development of each part of the system, with results shown in section 7. Conclusions about A.D.A developments are on Section 8.

2. Background

This section explores essential concepts to understanding and developing the proposed assistant, these concepts are Speech Recognition, Natural Language Processing, Text-to-speech Synthesis, and Microservices.

- **Speech Recognition**: can be interpreted as a matter of statistical optimization, finding the most probable sequence of words that a user might have said. Speech

recognition systems based on hidden Markov models (Figure 1) are usually adopted to figure out speech sentences. Its performance is mainly evaluated with Word Error Rate (WER), defined by the string distance between expected words and decoded ones. Given a similar scope, [Coucke et al. 2018] guided the approach employed to do such a task;

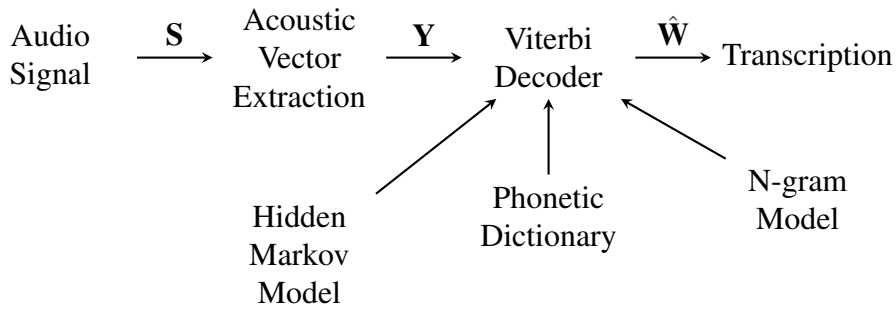

Figure 1. ***Speech Recognition Architecture based on Hidden Markov Model.***

- **Natural Language Processing:** is the study of techniques to translate natural languages, such as portuguese, to formal languages that machines are able to understand. Research in ways of representing word meaning resulted in deep embedding architectures, such as ELMo (Embedding for Language Models) [Peters et al. 2018], and generic language model architectures from large data sets, such as OpenAi Transformer (Open Artificial Intelligence Transformer) [Vaswani et al. 2017] and ULMFiT (Universal Language Model Fine-Tuning for Text Classification) [Howard and Ruder 2018]. All these models obtained trough deep learning neural networks have in common associated complex components (encoders, decoders, LSTMs), usually dealing with large data sets to perform several tasks including: language inference, intent analysis, machine translation. Currently, very few works do translation from natural language in brazilian portuguese to formal language. [Roman 2001] is one that does this based on the works of [Grosz and Sidner 1986] along multi-agent theory. [Luz 2019] based on LSTM encoder-decoder network architecture translates from natural language to SPARQL, an data retrieval formal language;
- **Text-to-speech Synthesis:** speech synthesis is a fundamental piece in the voice interaction loop between user and machine in a virtual assistant, as it is responsible for providing appropriate user feedback, given a certain uttering. Knowledge about processing and manipulating this interface is required in order to adapt or even propose new voice interaction models. To obtain a satisfactory voice interaction, an end-to-end architecture is required, one that integrates and trains all its models together. However, current open-source solutions do not present information in relation to performance, capacity and necessary resources for comparison;
- **Microservices:** is currently one of the main approaches for distributed software development, and it is based on two main characteristics: loose coupling and high cohesion [Newman 2015]. Microservices are defined as an architectural style that "structures an application as a set of loosely coupled services", which implements business capabilities [Richardson 2018]. Some of the main benefits offered by this architectural style are scalability, easy deployment, technological heterogeneity,

resilience, replaceability, and flexible and organized teams. A reactive microservice architecture is one based on the isolation and asynchronous communication of services, to provide greater software resilience and elasticity [Bonér 2016].

3. A.D.A – Advanced Distributed Assistant

The project A.D.A. aims to build an open source personal distributed intelligent assistant, that understands and processes Portuguese voice commands and allows user interaction with several IoT devices. A.D.A. is designed as a microservice architecture, where each microservice is responsible for part of the processing flux.

In order to illustrate A.D.A.'s architecture, Figure 2 presents the components that are being developed along with the data flow among them. These components are implemented according to a reactive microservice architecture [Bonér 2016], in which services are independent and communicate by brokers according to a publish-subscribe mechanism. Each subset of services and the architecture implementation itself is being researched and developed in a scientific initiation. The six modules are:

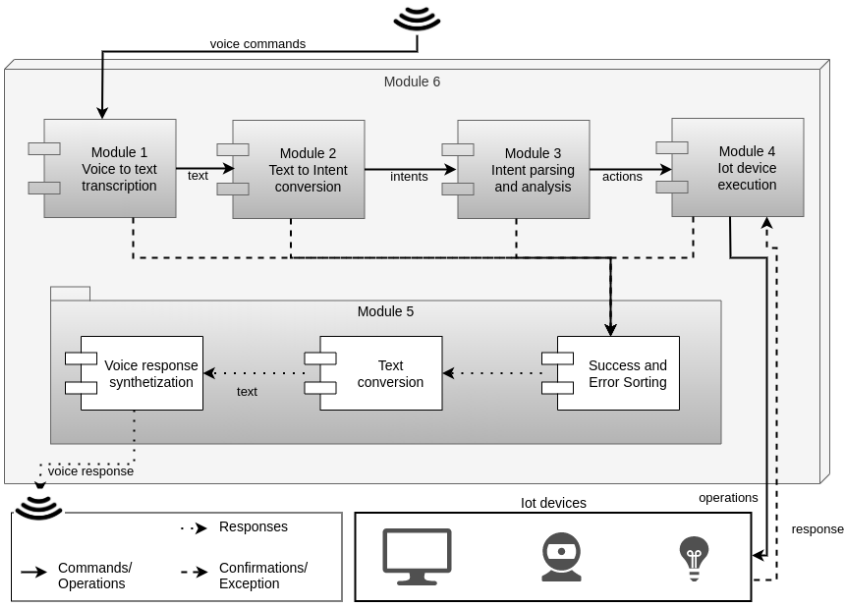

Figure 2. ***A.D.A. Architecture***

- **Voice to text transcription**: This subproject aims to implement voice-based user interaction on A.D.A. by processing sound signals and transcribing sound to natural language text [Abdul-Kader and Woods 2015]. This module must recognize a voice command, initiated by wake words, and transcribe user voice commands into simple syntactic structure sentences;
- **Natural Language Understanding**: This subproject aims to study several Natural Language Understanding techniques [Tur and Deng 2011] in order to develop software capable of extracting intent and objects of user imputed commands. This module must extract valid intents from input texts or recognize unclear sentences;
- **Intent Execution**: This subproject aims to implement an interpreter [Gamma 1995] capable of analysing user intent, alongside command context, and to issue actions to available IoT devices connected to an A.D.A. instance. The interpreter should choose appropriated sequence of executions and report any failures or success;

- **Device Connection**: This subproject aims to design IoT connections to enhance capabilities of A.D.A. This module must manage all IoT connections and provide an API for sensor reading and actuation controlling and also receive and send data to remote servers executing other A.D.A. modules. This module must allow device connection for operation execution and measurements collection requests;
- **Text-to-speech synthesis**: This subproject aims to study synthetic voice design and develop a subsystem capable of generating human-like voice to respond the user with sound executions. It will report commands execution success or failures in any other module back to the user;
- **Operations Infrastructure**: This subproject aims to study microservices architectures communication and organization strategies, in order to integrate other A.D.A. modules and provide a cloud native execution environment while not binding execution with specific cloud providers or other execution environments. It should focus on how to develop integration tests, enhance system fault tolerance and scalability [Newman 2015].

3.1. Usage Flow

A.D.A. should recognize a user speaking in order to execute his commands. It must capture the sound, using a microphone and process this audio file to transcribe from speech to text [Module 1]. The next step is identification of the capture sentence syntactic structure and to find commands emitted, generating an intent of device actions execution [Module 2]. The operations must be executed in order to best answer user expectations, leveraging context and devices connected to A.D.A. [Module 3]. Throughout execution of commands, success or exceptions in execution are reported back to the system by IoT devices [Module 4]. A.D.A. process any exceptions and successes back to the user by synthetic speech [Module 5]. All services as connected in the same infrastructure with previously defined communication patterns, allowing remote or local service allocation [Module 6].

3.2. System Development

As A.D.A. has a parallel development process among a set of sub projects, some measures are needed to synchronize its development. Specially since services transmit distinct kinds of data between each other and should be integrated in production stage. To ensure that services fulfill its requirements, we used contract design in development, where each service defines its inputs and outputs model and implements tests assuring its output is similar to other services input expectations [Crispin and Gregory 2009].

After understanding the fundamental concepts related to this research and how our assistant works, the next section explores some challenges associated with the development of virtual assistants.

4. Challenges

Voice based interfaces were historically considered a secondary interaction method. However, considering the increasing number of voice based assistants, this kind of interface stopped being an auxiliary resource or restrict to specific fields to become an instant task delegation mechanism [Reichenspurner et al. 1999, Pradhan et al. 2018]. This change in approach led the community to look closer at systems of this nature, researching in order to identify existing challenges in voice based assistants development and improvements in user experience [Myers et al. 2018]. Among current known challenges, some are:

- **Vague expression translating and intent identification**, since interpretation of poorly understood commands, without previous configuration, could possibly depreciate user experience [Rong et al. 2017, Myers et al. 2018];
- **Absence of adequate complex task assistant**, such as long duration tasks, in which a sequence of steps must be accomplished in a specific order, requiting a conversational interface [Vtyurina and Fourney 2018, Porcheron et al. 2018];
- **Absence of synthetic voice expressiveness** impairing user conversation experience and possibly its feedback understanding, once intonation, pauses, tonality and linguistic accents are also part of verbal communication [Fiannaca et al. 2018];
- **Automatic Scalability**, since an increase in the number of users will lead to poor performance if the execution environment and system architecture can not be horizontally scaled in order to deal with incresing connections without wasting resources [Cardellini et al. 1999];
- **Extensibility** in order for the system to be easily upgradable and receive new connections from new devices, new kinds of devices and new services to be incorporated in the command processing workflow [Newman 2015];
- **Context and dominion Recognition** in order to correctly select user intent without user constant consultation [Springer and Cramer 2018], correctly assessing user position and device position as a method of object identification in user command input, as an example.

5. Related Works

Current virtual assistants market is dominated by big tech companies [Felix Richter 2016], with products aimed primarily towards english speaking countries, while portuguese support only added later. Open source solutions are still rare, despite accelerated growth predticions for the near future [Ronan de Renesse 2017], keeping the common user locked on commercial closed source assitants that collect their data to improve themselfes.

5.1. Proprietary Assistants

Closed source assistants are described as "black boxes", since they offer input methods (voice and API) and output (sound and screen) while not disclosing any operations in between. They offer more advanced interaction and performance, specially in regards to voice-to-text and text-to-voice processing. However, user privacy is unclear and device portability is lacking, since these tools usually are bound to specific devices or systems.

Among commercial assistants, most notable are: Google Assistant (Google), Siri (Apple), Alexa (Amazon)[1], Cortana (Microsoft), and Bixby (Samsung)[2]. Other successful assistants not English-speaking centric include: Xiaowei (Tencent), AliGenie (Alibaba Group), Baidu Duer (Baidu), and Alice (Yandex)[3].

5.2. Open Source Assistants

Open-sourced assistant disclose, in part or all, its source code and architecture, allowing community-driven development and feature improvement. Despite open access, these

[1] `assistant.google.com`,`apple.com/siri/`, `developer.amazon.com/alexa`
[2] `microsoft.com/pt-br/windows/cortana`,`samsung.com/br/apps/bixby`
[3] `xiaowei.qcloud.com`, `iap.aligenie.com`, `dueros.baidu.com`, `yandex.ru/alice`

assistants may have purchasing costs on their user part. Also, their performance and user experience are worse when compared with commercial solutions.

Although there exists a reasonable number of open source NLI (Natural Language Interface) implementations, end-to-end solutions, which include voice recognition, intent parsing and voice feedback are still rare. Two notable exceptions are Mycroft and Snips[4], open source end-to-end agents, which are not completed yet.

Several open source assistants use propietary tools in order to offer an end-to-end user interface, with Saiy[5] as an example. It uses voice-to-text and text-to-voice of its host operationg system. Nonetheless this being a more pratical and providing improved user experience, it eliminates user privacy guarantees.

5.3. Virtual Assistants Comparision

As Table 1 shows, only commercial assistants provide full support to Portuguese speaking, while only Mycroft proving some support among open source ones. Most of them can be accessed by multiple devices, with some commercial ones bound to specific devices (Bixby) or operating systems (Siri and Cortana). Full dialogue support is only found on three assistant (Alexa, Bixby, Google and Siri). IoT Integration is common among them, only lacking on Bixby and Cortana. All commercial assistants and Mycroft do not enable local data processing, Snips being the only one with this feature. Out of the open source assistants, none are natively able to distribute its processing between remote and local environments. A.D.A. is design to be an intelligent system able to perform this processing distribution.

	Assistants							
Functionalities	Bixby	Cortana	Siri	Alexa	Google	Mycroft	Snips	ADA
Supports pt-BR	x	x	x	x	x	∼		x
Dialogue	x		x	x	x			x
Multi-device		x	x	x	x	x		x
IoT Integration			x	x	x	x	x	x
Multi-plataform				x	x	x		x
Open Source						x	x	x
Local Execution							x	x
Distributed Env.								x

Table 1. ***Project proposal and current assistants functionality comparison.***

After understanding some development challenges and conducting a comparative analysis of different virtual assistants with A.D.A., in the next section we present the methodology adopted for the development of this research.

6. Methodology

Acoustic Models for speech recognition, such as Hidden Markov Model, has its parameters estimated through a training dataset build from audio files and speech transcription

[4] `mycroft.ai, snips.ai`
[5] `saiy.ai`

from each of those files. There are little to none available open-source data sets of this kind in the Portuguese language, where the narrator speaks like a regular citizen. To solve this problem, we chose to build a data set by extracting audio from videos with written subtitles on YouTube[6].

An HMM-based system architecture was built based on open-source software and available systems. The Kaldi project [Povey et al. 2011] provided most of the tools needed for system development. The acoustic model was trained solely on data extracted from YouTube, thus dealing with the lack of labeled audio in Portuguese. Pre-existing components—to be later adapted or created—such as a lexicon [Ashby et al. 2012] and a language model [Batista et al. 2018] were used at first.

The purpose of the NLP module is to understand the natural language command from the previous module and translate it to a formal compilable language. In this way, we adapt the system proposed by [Roman 2001], which is based on the attentional, intentional and linguistic structures[Grosz and Sidner 1986], in addition with the task structure, that aims to map the domain dependency.

Concerning user interaction, we investigated the current state of the art of end-to-end speech synthesis to select the best architecture that could be able to synthesize natural voice in Brazilian Portuguese. Finally, five were reviewed: Char2Wav [Sotelo et al. 2017], ClariNet [Ping et al. 2019], Deep Voice 3 [Ping et al. 2018], Tacotron [Wang et al. 2017], and Tacotron 2 [Shen et al. 2018]. Based on criteria such as source code availability, the vocoder class, data pre-processing requirements, network structure, and Mean Opinion Score (MOS) values, we selected the Taconton 2 as the most fitting architecture.

To support the A.D.A infrastructure, the research method adopted is composed of exploratory bibliographic research on microservices and one case study to integrate different parts that compose the assistant. Since this architectural style allows to independently and concurrently develop each of the different services, by separated and specialized teams, with different tools and technologies.

For implantation of microservices, we used environment isolation tools for service execution, which are one of the best practices in microservices development, because it separates execution environments, facilitating technology heterogeneity with no impact on other services development. . We chose Docker[7] containers for environment isolation because of its large support on the open-source community. For container orchestration, we decided to use Kubernetes[8], since it allows container instantiation on distinct clouds or other environments, and it provides a higher number of features between all orchestration solutions [Truyen et al. 2019].

7. Results

Choosing to use microservices for designing A.D.A. allowed each undergraduate student to code individually and have its own source code repository. This led to a distinct pace in student productivity, with some initiations advancing more than others, especially the machine learning related ones. The source of this difference in productivity can be at-

[6] youtube.com
[7] docker.com
[8] kubernetes.io

tributed to the level of theoretical and practical dependence of some scientific initiations from other ones, even while being developed separately. Despite being necessary for a functional assistant, the infrastructure and IoT initiations were not fully developed because they require other modules to be already functional. The adoption of contract based development also was difficult, since students had not enough knowledge to establish contracts between all modules.

The project, on its current design, could be developed more efficiently by diving in development phases, with microservices dealing with the core of user interaction that was developed in a first phase and adjacent system and infrastructure developed in a second phase. Thus, in the remainder of this section, we present the main results achieved in each scientific initiation project.

In order to evaluate the speech recognition model, a test dataset containing 1175 audio clips was selected and separated from training dataset. This strategy was adopted due to computational limitations for executing a more robust evaluation, with cross-validation. The Word Error Rate (WER) between sentences decoded from this dataset was 44.3%, with wrong sentences rate being 87.2%.

Several NLP techniques and open source solutions were researched and compared. To translate from Portuguese to formal language, we considered using BERT [Devlin et al. 2018], which is considered one of the most advanced solutions in this field. BERT infers user intent with a very high success rate and can infer word meaning from its position in each sentence.

Our study on IoT, we researched some IoT connections and compared them is order to assess the best alternative for devices connected to A.D.A.. Some general IoT connection services were evaluated and we used a Raspberry Pi and a smart lamp to begin testing device connectivity. Also, we investigated methods of organizing command executions and how it might impact user experience.

For our voice synthesizing, in order to reduce the amount of data requirements in Brazilian Portuguese, as well as overall training time, we decided to train our Tacotron 2 network using a transfer learning method, which consisted on providing Portuguese data to a previously trained model in English language with the LJSpeech dataset[9]. We also evaluated the effect of domain adaptation using two datasets in Portuguese: one with multiple speakers and another with a single speaker.

Based on bibliographic research on microservices, we were able to justify its usage for designing the A.D.A. architecture. Therefore, we can claim that the microservices architectural style facilitates the implementation of requirements such as scalability, elasticity, resilience and extensibility. Furthermore, it favors technological heterogeneity, as well as the composition of more productive teams. Such aspects are very relevant in the context of this work, since A.D.A. is a project with components very distinct with each other, in their internal design and communication patterns.

8. Conclusion

The main objective of this work is to develop an intelligent distributed personal assistant, based on free software and hardware, that must understand and process voice commands

[9] `keithito.com/LJ-Speech-Dataset/` - last accessed in 2020

in Portuguese. Based on our research, the development of a distributed virtual assistant is a challenging undertaking because it involves research into several distinct computer science fields and constant coordination between all researchers.

At this moment, greater advances were accomplished in modules related to user interaction, regarding voice recognition and transcription, and voice synthesizing. Modules related to IoT and Infrastructure had fewer advancements due to their dependence on user iteration cycle already being built, so that they are incorporated, connected, and tested to their full scope. Our future goals include expanding A.D.A.'s capabilities within each module and by adding new modules to interact with other intelligent systems. Among the capabilities to be expanded are the support for more IoT devices and the offer of a range of synthesized voices for user selection.

References

Abdul-Kader, S. A. and Woods, J. (2015). Survey on chatbot design techniques in speech conversation systems. *International Journal of Advanced Computer Science and Applications*, 6(7).

Ashby, S., Barbosa, S., Brandão, S., Ferreira, J. P., Janssen, M., Silva, C., and Viaro, M. E. (2012). A rule based pronunciation generator and regional accent databank for portuguese. In *Thirteenth Annual Conference of the International Speech Communication Association*.

Batista, C. T., Dias, A. L., and Neto, N. C. S. (2018). Baseline acoustic models for brazilian portuguese using kaldi tools. In *IberSPEECH*, pages 77–81.

Bonér, J. (2016). *Reactive microservices architecture : design principles for distributed systems*. O'Reilly Media.

Cardellini, V., Colajanni, M., and Yu, P. (1999). Dynamic load balancing on Web-server systems. *IEEE Internet Computing*, 3(3):28–39.

Coucke, A., Saade, A., Ball, A., Bluche, T., Caulier, A., Leroy, D., Doumouro, C., Gisselbrecht, T., Caltagirone, F., Lavril, T., et al. (2018). Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*.

Crispin, L. and Gregory, J. (2009). *Agile testing: A practical guide for testers and agile teams*. Pearson Education.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Felix Richter (2016). Digital Assistants - Always at Your Service. last accessed on 02/06/20 - www.statista.com/chart/5621/users-of-virtual-digital-assistants/.

Fiannaca, A. J., Paradiso, A., Campbell, J., and Morris, M. R. (2018). Voicesetting. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*, pages 1–12, New York, New York, USA. ACM Press.

Gamma, E. (1995). *Design patterns : elements of reusable object-oriented software*. Addison-Wesley.

Grosz, B. J. and Sidner, C. L. (1986). Attention, intentions, and the structure of discourse. *Computational linguistics*, 12(3):175–204.

Howard, J. and Ruder, S. (2018). Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.

Lovelace, A. K. and Toole, B. A. B. A. (1992). *Ada, the enchantress of numbers : a selection from the letters of Lord Byron's daughter and her description of the first computer*. Strawberry Press.

Luz, F. F. (2019). *Deep neural semantic parsing: translating from natural language into SPARQL*. PhD thesis, Universidade de São Paulo.

Mitchell, T. M., Caruana, R., Freitag, D., McDermott, J., and Zabowski, D. (1994). Experience with a learning personal assistant. *Communications of the ACM*, 37(7):80–91.

Myers, C., Furqan, A., Nebolsky, J., Caro, K., and Zhu, J. (2018). Patterns for How Users Overcome Obstacles in Voice User Interfaces. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*, pages 1–7, New York, New York, USA. ACM Press.

Newman, S. (2015). *Building Microservices: Designing Fine-Grained Systems*. O'Reilly Media, Sebastopol - USA.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Ping, W., Peng, K., and Chen, J. (2019). Clarinet: Parallel wave generation in end-to-end text-to-speech. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Ping, W., Peng, K., Gibiansky, A., Arik, S. Ö., Kannan, A., Narang, S., Raiman, J., and Miller, J. (2018). Deep voice 3: Scaling text-to-speech with convolutional sequence learning. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Porcheron, M., Fischer, J. E., Reeves, S., and Sharples, S. (2018). Voice Interfaces in Everyday Life. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*, pages 1–12, New York, New York, USA. ACM Press.

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., et al. (2011). The kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society.

Pradhan, A., Mehta, K., and Findlater, L. (2018). "Accessibility Came by Accident". In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*, pages 1–13, New York, New York, USA. ACM Press.

Reichenspurner, H., Damiano, R. J., Mack, M., Boehm, D. H., Gulbins, H., Detter, C., Meiser, B., Ellgass, R., and Reichart, B. (1999). Use of the voice-controlled and computer-assisted surgical system zeus for endoscopic coronary artery bypass grafting. *The Journal of Thoracic and Cardiovascular Surgery*, 118(1):11–16.

Richardson, C. (2018). *Microservice Patterns.* Manning Pubns Co.

Roman, N. T. (2001). Estudo de dialogos orientados a tarefa usando a teoria de multiagentes. Master's thesis, Universidade Estadual de Campinas, São Paulo, Brazil.

Ronan de Renesse (2017). Virtual digital assistants to overtake world population by 2021.

Rong, X., Fourney, A., Brewer, R. N., Morris, M. R., and Bennett, P. N. (2017). Managing Uncertainty in Time Expressions for Virtual Assistants. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems - CHI '17*, pages 568–579, New York, New York, USA. ACM Press.

Shen, J., Pang, R., Weiss, R. J., Schuster, M., Jaitly, N., Yang, Z., Chen, Z., Zhang, Y., Wang, Y., Ryan, R., Saurous, R. A., Agiomyrgiannakis, Y., and Wu, Y. (2018). Natural TTS synthesis by conditioning wavenet on MEL spectrogram predictions. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15-20, 2018*, pages 4779–4783. IEEE.

Sotelo, J., Mehri, S., Kumar, K., Santos, J. F., Kastner, K., Courville, A. C., and Bengio, Y. (2017). Char2wav: End-to-end speech synthesis. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net.

Springer, A. and Cramer, H. (2018). "Play PRBLMS". In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*, pages 1–13, New York, New York, USA. ACM Press.

Tankard, C. (2016). What the GDPR means for businesses. *Network Security*, 2016(6):5–8.

Truyen, E., Landuyt, D. V., Preuveneers, D., Lagaisse, B., and Joosen, W. (2019). A comprehensive feature comparison study of open-source container orchestration frameworks. *Applied Sciences*, 9(5):931.

Tur, G. and Deng, L. (2011). Intent Determination and Spoken Utterance Classification. In *Spoken Language Understanding*, pages 93–118. JohnWiley&Sons,Ltd.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Vtyurina, A. and Fourney, A. (2018). Exploring the Role of Conversational Cues in Guided Task Support with Virtual Assistants. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*, pages 1–7, New York, New York, USA. ACM Press.

Wang, Y., Skerry-Ryan, R. J., Stanton, D., Wu, Y., Weiss, R. J., Jaitly, N., Yang, Z., Xiao, Y., Chen, Z., Bengio, S., Le, Q. V., Agiomyrgiannakis, Y., Clark, R., and Saurous, R. A. (2017). Tacotron: Towards end-to-end speech synthesis. In Lacerda, F., editor, *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017*, pages 4006–4010. ISCA.

Weiser, M. and Mark (1993). Some computer science issues in ubiquitous computing. *Communications of the ACM*, 36(7):75–84.