electronics

Article

EdgeAvatar: An Edge Computing System for Building Virtual Beings

Neftali Watkinson, Fedor Zaitsev, Aniket Shivam, Michael Demirev, Mike Heddes, Tony Givargis *, Alexandru Nicolau and Alexander Veidenbaum

Donald Bren School of Information and Computer Science, University of California, Irvine Irvine, CA 92617, USA; watkinso@uci.edu (N.W.); fzaytsev@uci.edu (F.Z.); aniketsh@uci.edu (A.S.); mdemirev@uci.edu (M.D.); mheddes@uci.edu (M.H.); nicolau@ics.uci.edu (A.N.); alexv@ics.uci.edu (A.V.)

* Correspondence: givargis@uci.edu

Abstract: Dialogue systems, also known as conversational agents, are computing systems that use algorithms for speech and language processing to engage in conversation with humans or other conversation-capable systems. A chatbot is a conversational agent that has, as its primary goal, to maximize the length of the conversation without any specific targeted task. When a chatbot is embellished with an artistic approach that is meant to evoke an emotional response, then it is called a virtual being. On the other hand, conversational agents that interact with the physical world require the use of specialized hardware to sense and process captured information. In this article we describe EdgeAvatar, a system based on Edge Computing principles for the creation of virtual beings. The objective of the EdgeAvatar system is to provide a streamlined and modular framework for virtual being applications that are to be deployed in public settings. We also present two implementations that use EdgeAvatar and are inspired by historical figures to interact with visitors of the Venice Biennale 2019. EdgeAvatar can adapt to fit different approaches for AI powered conversations.

Keywords: edge computing; chatbot; artificial intelligence; art; virtual being

check for updates

Citation: Watkinson, N.; Zaitsev, F.; Shivam, A.; Demirev, M.; Heddes, M.; Givargis, T.; Nicolau, A.; Veidenbaum, A. EdgeAvatar: An Edge Computing System for Building Virtual Beings. *Electronics* **2021**, *10*, 229. https://doi.org/10.3390/electronics10030229

Academic Editor: Pietro Manzoni
Received: 30 December 2020
Accepted: 18 January 2021
Published: 20 January 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Ever since Alan Turing published his article titled "Computing Machinery and Intelligence" [1], there has been a coordinated effort by the computing community to generate systems that can have human-like conversations. These are known as dialogue systems and they generally involve several sub-fields of computer science, linguistics, and cognitive science, making them a true interdisciplinary product.

Dialogue systems can have different objectives. For example, the typical virtual assistant found in most of today's smart devices are task-oriented systems, conversational agents that interpret human instructions with specific goals. On the other hand, an agent with the sole objective of lengthening a conversation, without any specific task in mind, is known as a chatbot [2]. With the popularity of artificial intelligence, chatbots have become more robust and are commercially available in several disciplines [3]. If the chatbot has artistic elements and tries to emulate a personality to evoke an emotional response from the user, it is considered a virtual being. Virtual beings use all kinds of algorithmic approaches and sensory stimulus to converse with users and simulate the presence of an actual human behind such interaction. The term is new [4] and meant to separate the artistic aspect of these systems from other chatbots. For successful deployment in public settings, it is essential to create robust, flexible and immersive virtual beings that allow for artistic freedom and expressiveness.

1.1. Edge Computing

Recent technological advances and availability of computational resources in embedded devices have given rise to a novel paradigm called Edge Computing. The idea behind

Edge Computing is to bring data processing to where the data originates, i.e., at the edge of the network. With the availability of artificial intelligence dedicated processors and high speed Internet, we can design layered processing where critical tasks are prioritized and compute intensive data analysis tasks are offloaded to the middle layer or to the Cloud where additional resources are available. One of the important decisions when designing an Edge Computing network is to allow for a middle-ware to handle communication with the Cloud and to allow for a level of independence among edge devices that don't depend on Cloud services [5]. Figure 1 shows a diagram of a typical Edge Computing layout and topology. Dialogue systems usually include compute intensive tasks that are ideal for middle-ware offloading that would allow for a more natural workflow.

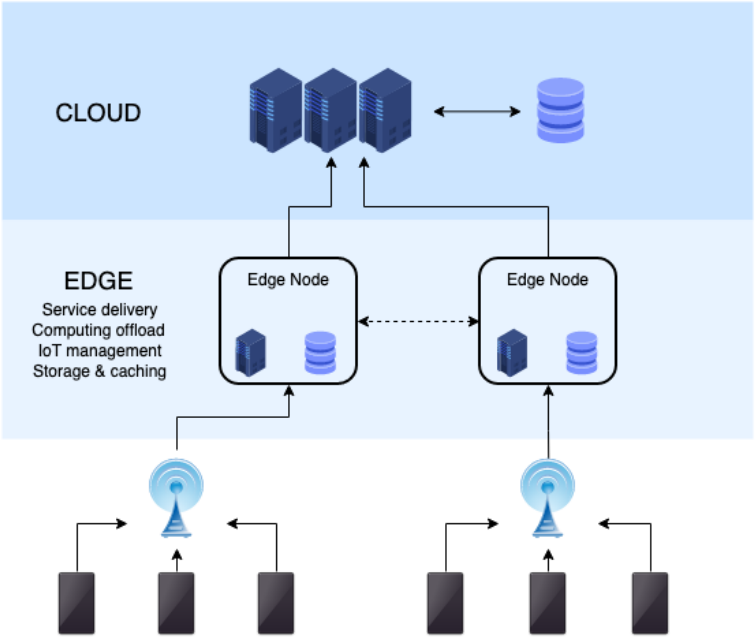


Figure 1. Abstract diagram representing the different layers in Edge Computing. Optimally, the higher layers deal with non-critical tasks that are compute intensive, which allows for a more resilient system. Source: Wikipedia

1.2. Contribution

In this article, we describe EdgeAvatar, a system that we designed with the intention of implementing it on two virtual beings that were on display at the Venice Art Biennale in 2019. In order to make the two systems work at an outdoor setting and create an immersive experience, we relied on principles of Edge Computing. While the two systems are similar in construction, they do show sufficient variability in behavior in order to illustrate the ability of EdgeAvatar to incorporate different technologies used for dialogue systems. For instance, the expensive task of transcribing audio to text and vice versa is offloaded to a separate device or process that doesn't intervene with other tasks of the system. We also delve into the aesthetics of the systems and special considerations regarding human privacy that were relevant for our design. These two systems used voice interaction and a virtual display to represent historic characters that would converse with visitors. One of the two systems was modeled after late Romanian leader, Nicolae Ceausescu, and the other system was inspired by the poet Paul Celan. The main goal of EdgeAvatar is to provide a common framework for Virtual Beings that allows artistic input to merge with modern technology. In this sense, we discuss how EdgeAvatar improves the incorporation of the artistic vision without compromising the technology behind it. This is thanks to a modular design that allows for simultaneous development of the different elements comprising the virtual agent. Since conversation is the core element driving the interaction, we analyze how decisions made during specific implementations of EdgeAvatar helped in creating an enjoyable and fluid experience. Finally, we describe the future direction of this work as well as a third system that was deployed as a mobile application. For each system, we describe the key elements and decision-making behind their design as well as list the major challenges in creating a successful immersive experience.

1.3. Related Work

There are several systems and applications for implementing dialogue systems and chatbots in public settings. We focus mainly on those that could be classified as Virtual Beings, that is, chatbots that have an artistic purpose for their interactions. Other works have tried to present general guidelines on how to create systems that provide users with an immersive experience [6]. For art and theater, the use of story based chatbots has seen the biggest effort in research. In the work of Hoffman et al. [7], a puppeteering system for robotic actors is presented. The research delves mainly into the physical device and how it can be programmed and controlled for acting sequences. Most recently, the research of Bushan et al. [8] proposes a system that uses an Edge Computing network to enable a story based chatbot that can interact with guests through sensing within a controlled space. Additionally, Vassos et al. [9] explores the use of a chatbot as a museum guide that uses a scripted story to inform users about the elements found in the museum.

While there are similarities between these systems and EdgeAvatar, it is important to note that, while they are also classified as virtual beings, these are story oriented systems. That means that the artistic representation is mainly embedded in the story that the chatbots are trying to tell or involve the audience in. This usually means that there's a controlled space where multiple sensors and actuators interact with each other. EdgeAvatar doesn't presume such control over the space nor is it constrained by a specific technology or story to be told. In fact, the applications we designed using EdgeAvatar are meant to participate in open-ended conversations. They could, however, be programmed to guide the audience through a story, but the main contribution of EdgeAvatar is the ability to adapt to what the artist is trying to communicate.

Regarding the technology supporting EdgeAvatar, using a cloud based service, such as a knowledge database, has been a main interest for creating chatbots that deal in domains where information is ever changing or too broad to be kept in a local server. The work of Chung et al. [10] explores the use of a cloud based database for a chatbot framework designed for healthcare. Additionally, the introduction of 5G networks has allowed for wireless, on-site applications where users can interact with chatbots maintaining quality of service (QoS). The work of Koumaras et al. [11] tests different commercial chatbots that rely on a cloud service to function, under extreme conditions (festivals, traffic jams, busy cafes) and found that, as long as connectivity was not interrupted, the applications would function properly with an average delay of 3 s per step, or 6 s as the maximum delay.

2. Materials and Methods

EdgeAvatar uses existing technology that is available for speech recognition and human computer interaction. The two main challenges we solve are: establishing a common framework for the different implementations and incorporating the aesthetic elements into our design.

2.1. Common Framework

There are three major technical sub fields of computer science that EdgeAvatar relies on: Speech Recognition, Information Retrieval, and Speech Synthesis. There are several solutions for each of these. However, for EdgeAvatar, we are interested in the interaction between each of the technical sub fields under specific circumstances. Since the inspiration for our system is human-like conversation, we divide the components of EdgeAvatar following the key components of the human's face anatomy. Therefore, the main modules of EdgeAvatar are divided as follows:

- Speech-to-Text (ears): This module's main objective is to receive vocal information from an interviewer and transcribe it into digital text. The text is later processed by the answer generation module.
- Subject Sensing (eyes): EdgeAvatar is designed for active interaction with passersby; therefore, it needs to sense whether a subject is in front of it to start an interaction. This problem is solved by using active object detection capable of sensing people.
- Answer Generation (brain): At the core of EdgeAvatar is the answer generation module. It takes a text query as input and generates an answer also in text form. The module needs to reflect the specific information of the character being emulated by EdgeAvatar; this is achieved in multiple ways as described in a later subsection.
- Visual Output (face): This is what the human participant sees and it is a visual representation or puppet. The face is always engaged and may involve dead time animations for when there's no interaction. This module is the most influenced by the aesthetic input and artistic inspiration.
- Text-to-Speech (voice): Just as the ears transcribe the vocal queries, the Text-to-Speech module converts the text back to an audio output that can be understood by the interviewer.

Figure 2 serves as a reference for the design behind EdgeAvatar. The flow of information starts with a vocal query from the interviewer, which is recorded and processed by the Speech-to-Text (STT) module, forwarded as text to the Answer Generation module. The answer is then processed by the Text-to-Speech (TTS) module which queues the display to show a talking animation while playing the audio component of the answer.

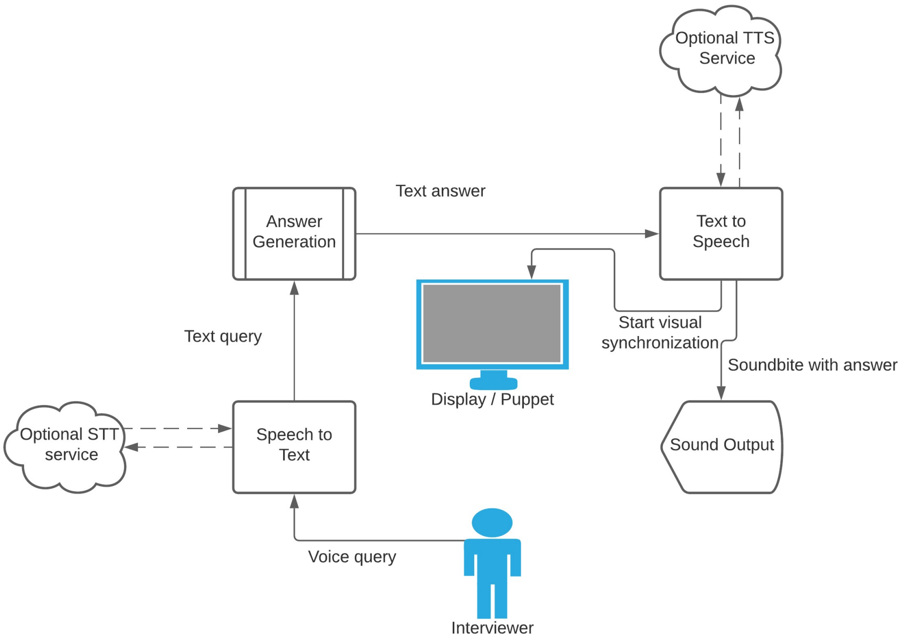


Figure 2. The general design of our systems follows a modular structure where each task is handled by a separate device or software. Alternatively, some modules such as the text to speech and speech to text modules could work as middle-ware for Cloud services.

One of the reasons for using a modular design is so that each component can be easily replaced with different technologies. For example, the diagram in Figure 2 shows an optional connection to Cloud services for Speech-to-Text (STT) and Text-to-Speech (TTS). When using such services, these modules should be implemented as middle-ware that handle communication with the Cloud separately from the local network. The main reason for this is to handle issues such as data movement latency separately, without disrupting the overall workflow. In such cases, the STT and TTS modules are equivalent to the middle layer in Edge Computing.

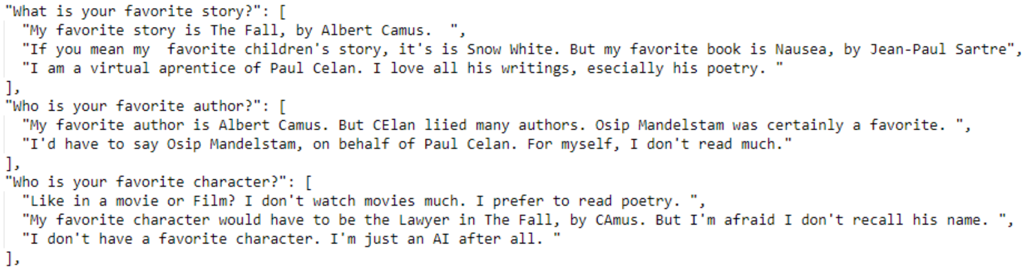
2.1.1. Listening and Sensing

It is important to note that the diagram for the core design doesn't include the continuous sensing module. This is because continuous sensing is not part of the main workflow and rather an optional addition that manages the STT module. Since constant listening for conversations is costly and oftentimes results in false triggering (i.e., reacting to a conversation that was not intended for the virtual being), continuous sensing allows the system to disable recording until needed. This management could be as simple as implementing a push-to-talk microphone or perform object detection to detect if someone is in front of the system. Ultimately, the desired behavior of the virtual being, the physical housing constraints, and the environmental limitations play a key role in the decision-making process that determines how continuous sensing is to be done.

2.1.2. Brain

The system's *brain* is the module in charge of generating responses to input queries. While EdgeAvatar is meant for building virtual beings, the brain is in essence a chatbot. This means that we can use any approach to dialogue systems that is compatible with our input and choosing the one that fits the intended character the best. This includes term matching approaches all the way to sophisticated deep learning algorithms.

Term matching allows one to provide quick answers for the most common use case scenarios. This gives the interviewer a feel that the bot is responsive to their questions. This gives a more natural feel than relying solely on deep learning algorithms. To match a question with an answer, we first process it through the STT module. After transcribing the question, we remove all the punctuation from the question and normalize the text by applying stemming and lemmatization. This removes conjugation and unnecessary information for matching. After this step, we apply a TF-IDF algorithm [12] followed by a computing cosine similarity between the lemmatized question and all the answers in the database. We discard all the answers with a cosine similarity below a predetermined threshold (default is 0.70) and pick a random answer from the top 3 (if available). If we don't have at least one answer above the threshold, then we generate an answer using a recurrent neural network or randomly select one from a list of *filler* answers. Figure 3 illustrates how questions and answers are stored in the database. Each question can have different versions to improve matching, and some questions have several answers to choose from. It is important for the interviewer's experience that the interaction is not repetitive nor monotonous, so, even if the question is factual, the system has different answers to choose from.



```
"What is your favorite story?": [
  "My favorite story is The Fall, by Albert Camus.  ",
  "If you mean my  favorite children's story, it's is Snow White. But my favorite book is Nausea, by Jean-Paul Sartre",
  "I am a virtual aprentice of Paul Celan. I love all his writings, esecially his poetry. "
],
"Who is your favorite author?": [
  "My favorite author is Albert Camus. But CElan liied many authors. Osip Mandelstam was certainly a favorite. ",
  "I'd have to say Osip Mandelstam, on behalf of Paul Celan. For myself, I don't read much."
],
"Who is your favorite character?": [
  "Like in a movie or Film? I don't watch movies much. I prefer to read poetry. ",
  "My favorite character would have to be the Lawyer in The Fall, by CAmus. But I'm afraid I don't recall his name. ",
  "I don't have a favorite character. I'm just an AI after all. "
],
```

Figure 3. Response format as it is saved in the database. Each question has more than one option to use as an answer. Not shown here is that questions also had variations for matching

2.2. *Conversation Simulation*

A key element for the functionality of chatbots and virtual beings is the conversation itself. Since EdgeAvatar can incorporate any state-of-the-art approaches to conversation generation, we are mostly interested in the fluidity of it. That is, given EdgeAvatar's infrastructure, we want to guarantee that QoS is good enough to communicate the feeling of a natural conversation. For this, we need to establish an acceptable delay range for answering the interviewers' question without interrupting them. In determining the lower bound, we found that generating answers too quickly (10 to 200 msec after a pause from the interviewer) would often be perceived by test subjects as too reactive and artificial. Therefore, we implemented a *waiting* range so that, if the system is ready to answer before the lower end of the range, we delay the interaction. If the answer is above the range, then we use random *filler* answers to simulate a thought process. To determine the lower bound for this range, we looked into research from the field of cognitive science regarding how fast humans respond in conversations. The work of Tovée et al. [13] reveals that humans generate new thoughts at the speed of 300–500 msec, while Magyari et al. [14] note that, when engaged in a conversation, early anticipation can allow a human to generate answers as fast as 200 msec, which is the delay we chose for the lower bound to our *waiting* range. We based the upper bound on findings form research applied specifically to dialogue systems. Roddy et al. [15] and Maier et al. [16] explore the use of Long-Term Short-Term Memory networks (LSTM) to determine when is the robot's turn to answer. They find that answering within the range of 1000–2000 msec (1 to 2 s) results in the lowest delay to interruption threshold. Applied to our system, if the answer is not ready after 2 s, we use a *filler* answer such as: "Interesting question, let me think about it", "Give me a second to answer that one", and "I will try to answer your question".

2.3. Behavior and Aesthetics

EdgeAvatar was designed for art exhibitions and it is therefore influenced by artistic input. The system was designed to allow for the ability to select from a number of advanced technologies in order to faithfully incorporate the artist's intentions on how the system behaves and appear to users at an exhibit. This level of design flexibility allows for the systems to become much more robust and work in many different forms or environments, maximizing accessibility.

Some modules are, therefore, influenced by the aesthetics. For example, the speech synthesis module (voice) could use pre-recorded messages that allow for the inclusion of voice acting prior to deployment. Alternatively, different services for TTS with increasing quality of speech can be easily implemented in this module. Advances in TTS have allowed for some near-human sounding services. The choice is almost entirely left to the artist.

No module is more influenced by art than the visual representation or the face of the system. Figure 4 shows the two styles that were chosen for the two systems described in the Results section. One uses a completely 3D rendered object built using digital design software and the other uses a static image that is distorted to appear as if it's talking. Talking can be done through lip-syncing, where the object uses coded rules to open and close the mouth along with the audio, or alternatively using a pre-recorded video which gives more control over the quality of the mouth movement. Based on artistic considerations, we make sure that the face is synchronized with the TTS (voice) module. Optionally, we add dead time animations; these are subtle movements sometimes accompanied by short speech for when the virtual being is not engaged with an interviewer or to invite someone to start a conversation.

On the technical side, EdgeAvatar is able to adapt to different challenges. For example, in a location with little or no Internet connection, it may instead be necessary to rely on an offline and local STT module as a backup for when the Cloud service is not available. Similarly, speech synthesis can be generated previous to the deployment for known, pre-written answers. Finally, speech recognition could be dropped completely when quality of service cannot be guaranteed and switch to textual input through a writing peripheral. A separate issue lies in deciding how many devices will be used. Certainly, each module can run on a separate device if sufficient number of devices is available. Alternatively, a single device that is capable of running parallel processes, can be used to deploy all modules. These are decisions that make the overall system robust, economical, and more energy efficient.

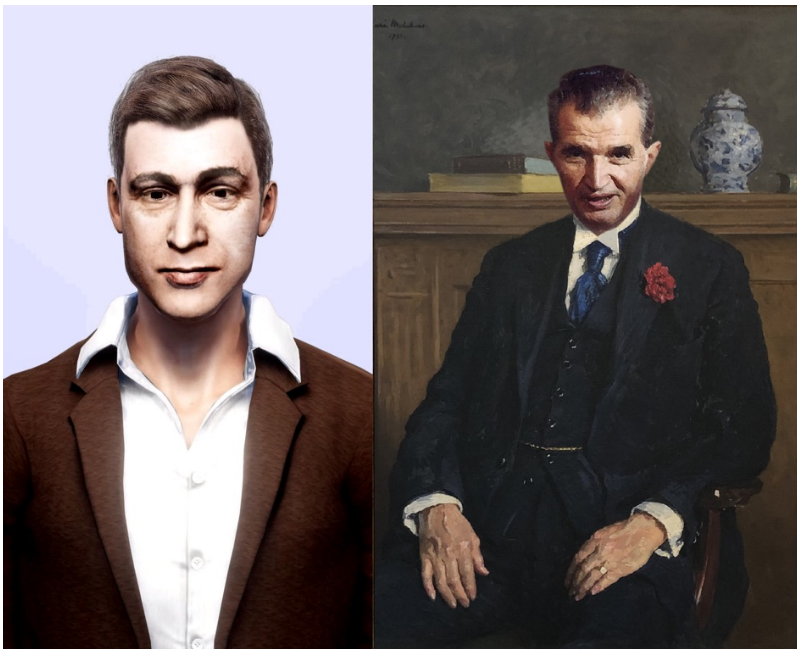


Figure 4. This is the visual interface for the systems based on Paul Celan (**left**) and Nicolae Ceausescu (**right**). Celan is a 3D rendering based on images from the poet, whereas Ceausescu uses image distortion to make an image appear as if it's talking.

3. Implementation

EdgeAvatar was used to create two virtual beings to interact with visitors at the Venice Art Biennale 2019. The Venice Art Biennale, or *La Biennale di Venezia*, is the oldest international exhibition for contemporary art [17]. While extending throughout the whole city of Venice, Italy, the formal event space is at the Giardini, where national pavilions house the most relevant artistic works from around the world. Each edition of the Biennale has a theme, for 2019, the theme was *May you live in interesting times* , and had a wide variety of art exhibits, some exploring the intersection of technology and art, and some more conventional. Along with artist Belu Fainaru [18], we installed our virtual beings at the Romanian Pavillion within the Giardini.

The Giardini della Biennale (the gardens of the Biennale) is a space with buildings that are over 130 years old [19]. Considering the cultural and historic heritage, it is impossible to make big changes to the infrastructure, including getting wired internet (see Figure 5). While it has access to simple commodities (water and electricity), and cellular reception is adequate during regular use, the bigger crowds such as those present during the Biennale can have a big impact on network stability. This meant that our systems had to optimize the use of a low-bandwidth Internet service and also be able to withstand the Venetian weather and remain fully functioning for over six months. During this time, the systems saw torrential rain, sun exposure under a hot and humid weather, among other natural phenomena that threatened the system's resilience. Additionally, the event had almost 600,000 visitors, and each system interacted with approximately 20 visitors per hour during peak operating times. This presented unique challenges to our design.

Each of the two virtual beings were built to represent a specific historic character: Poet Paul Celan and Romanian dictator Nicolae Ceausescu. Both of them share the same general framework from EdgeAvatar, though there are key differences that we describe below.

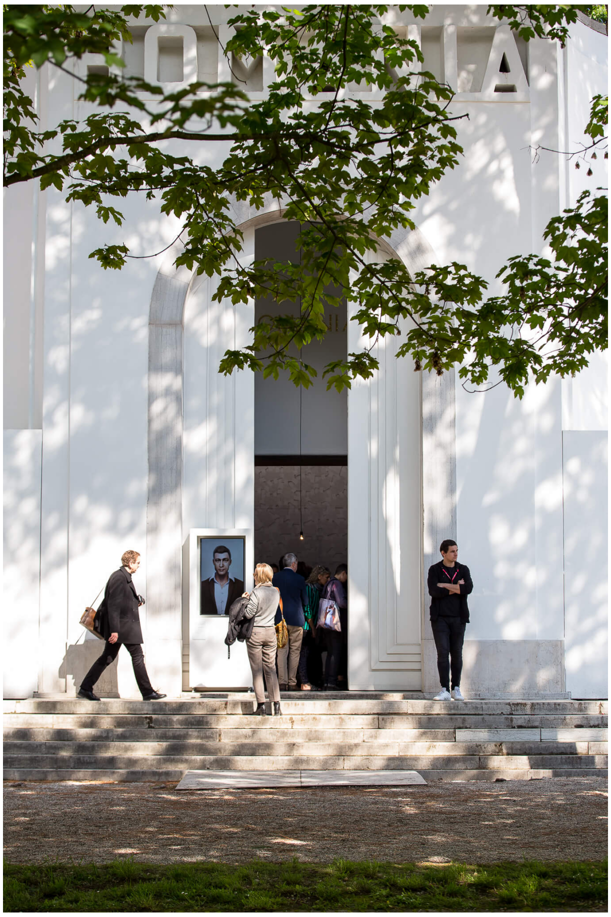


Figure 5. Installation of the system based on Paul Celan at the Romanian Pavilion in the Giardini della Biennale. As seen in this picture, the system had to withstand the Venetian weather and be bright enough to be displayed outside under the sunlight. Source: YAP studio

3.1. Celan

Paul Celan was a Romanian poet of Jewish heritage. He was a Holocaust survivor and spent most of his later years in France. His poetry style is dark and complex, heavy with symbolism. His most famous work, *Todesfuge*,is considered to be one of the most important poems of its time and one that captures the horrors of concentration camps like no others [20]. The language of his poems often share similarities with that of biblical excerpts. When interviewed, Celan would often stay away from straight answers, adding a mysterious element to his persona. The virtual being named Celan imitates this style of conversation, incorporating a style akin to Celan's work by incorporating common symbolism found in his poems. This was achieved through a fine-tuned adaptation of EdgeAvatar to fit the desired outcome.

Interaction Initiation

Like many artists, Paul Celan was rarely seen starting a conversation. He was more often the one being interviewed than the interviewer. Because of this, we didn't want virtual Celan to be eager to talk to people. Celan had a scripted behavior to entice people to interact with it instead of active sensing. Using pre-generated animation with internal timing had Celan looking around, sometimes reciting excerpts of his poems or lamenting that nobody would talk to him, but never actively inviting someone to talk. In order to minimize false triggers over conversations not intended for the system, we implemented a push-to-talk microphone to interact with the exhibit. The user who wanted to interact with Celan would pick up the microphone and push a button to start the conversation (see Figure 6).

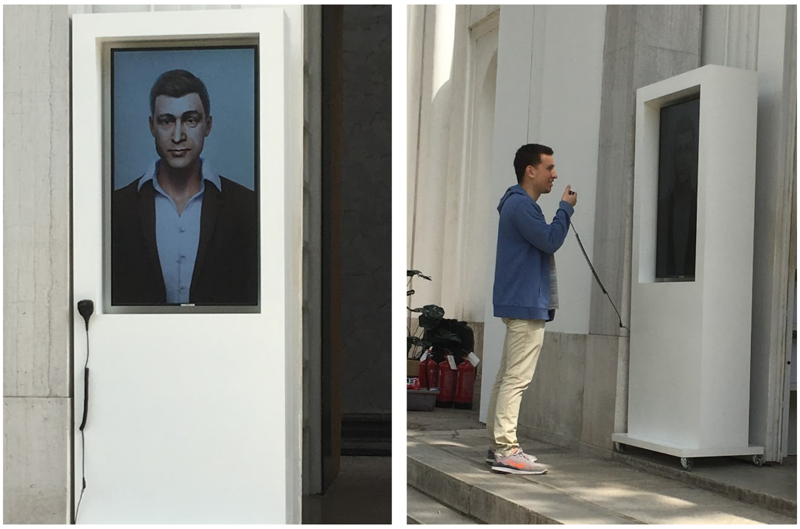


Figure 6. To the left of this figure is the final installation for the system inspired by Paul Celan. To the right one of us is demonstrating how to talk to the system.

3.2. *Recognizing Speech*

Once the interviewer starts talking to Celan, the challenge is to recognize their speech. The accuracy of the transcription determines the quality of the answer to the questions asked. In a setting like the Venice Biennale, the users are diverse in terms of their nationality, language, and culture. This creates the challenge for recognizing their words/sentences (spoken in English) with different pronunciations. We explored multiple STT services before making our final decision of settling on Microsoft Azure's Cognitive Services [21], which gave us the robustness needed for this setting. While there are solutions available to perform offline (processing on the system) STT, the Azure Cloud service provided the best performance, in terms of accuracy, of the speech translation to text.

The short turnaround between recording speech and its translation to words is very crucial to maintain conversations with the interviewer (i.e., the user). If the delay between the question and the answer is too long, then users would perceive that the system is not responding, eventually reducing their interest. The Azure Cloud service could also translate speech with high efficiency in terms of both the processing time and the cost incurred. During testing, we found that, even with inconsistent Internet speeds, the on-demand service with Azure was resilient enough. Once the text is received from the Azure Cloud service, we forward the text as the query or question from the interviewer to our Answering System.

3.2.1. Answering System

Since we don't have an extensive corpus of Celan interviews, we had to create a script simulating how the poet would answer questions. For factual questions regarding his background or some elements of his poetry, the system would answer with information gathered from his biography or published analyses of his work. For all other questions, we had a Recurring Neural Network (RNN) based on Char-RNN [22] that was trained on writings by Celan to generate answers on a character by character basis, making every answer unique. However, since his writings alone were not enough training data to generate coherent answers, we complemented the data using text from the Cornell Movie Dialogue Corpus [23]. This allows the network to learn proper sentence structure and how to interact in a conversation. Additionally, we chose texts that had similar language and symbolism as Celan's poems. Specifically, we used the old testament from the King James Bible and Shakespeare's plays and sonnets, the text was extracted from transcriptions made by the Gutenberg Project [24]. The network takes in a specific format that simulates conversation between two entities. While the dialogue corpus is already in this format, the Bible text was reformatted using each verse as a separate entry in a conversation.

For Shakespeare's work, we treated each line as an entry. We followed a similar formatting process for Celan's writings.

We combined the movie corpus, the Bible text and Shakespeare's works to train the network using the hyper-parameters described in Table 1. After this, we verified that the answers the network gave to our queries followed the right structure and were coherent. Then, we trained the network for 30 additional epochs but using only writings from Celan. In the end, we produced a network that could follow a conversation using language similar to Celan's poems. This network is meant to be used for answering questions that don't match to any of the factual answers in the database. However, due to computing limitations of the systems used in Venice, we generated the answers to common, non-factual questions prior to deployment (the latency of running the network live was too high for the computer we used). During live interactions, the system used a term frequency based matching algorithm to retrieve answers that were either scripted or previously generated from the RNN.

Table 1. Hyperparameters for the Recurrent Neural Network used to generate answers

Hyperparameter	Value
Layers	3
Model	Gated Recurrent Unit (GRU)
Sequence length	40
Epochs	100
Learning Rate	1×10^{-5}
Decay Rate	0.975

3.2.2. Speech Generation

When Celan is answering a question, it uses synthetic speech generation to translate the text answers into audio. Offline solutions that were supported by our system sounded unnatural or *robotic*. In turn, we decided to use Amazon Polly [25], an STT Cloud service to generate our answers. Since all the answers were generated prior to deployment, we could process the recordings prior to the event in order to minimize requests to the Cloud service.

3.2.3. Hardware

For Celan, we used a fanless, *rugged* computer that could withstand the weather. The system was equipped with the following:

- Casing: Neousys Nuvo-7160
- Processor: Intel Coffee Lake i7-8700
- Memory (RAM): 6 GB DDR3
- Storage: 500 GB SSD Hard Drive
- Graphics card: Nvidia GTX 1660 Ti Mini ITX OC 6G 192-bit GDDR6

The sound and image were played on a weatherproof, 55-inch television. Everything was mounted in a custom made cabinet.

3.2.4. Round-Trip Latency

As mentioned in a previous section, the total delay between asking a question and receiving an answer should be within the chosen range of 200 to 2000 msec. If retrieving an answer takes longer than 2 s, we play a single filler answer prior to the actual one. This gives the system an extra 2 s to respond. When using the Azure transcription service, the question is uploaded in real time at 256 Kbps; therefore, the latency is insignificant as long as the internet connection can support it. While running tests, as long as the upload speed is over 1 Mbps, we didn't observe any upload latency. Afterwards, the text is sent to the answer generation module. Receiving the text and generating an output (answer) adds 100–200 msec to the total delay. The final step is to find and play the audio file for that answer which adds 50–100 msec. Therefore, as long as there is a stable internet connection and the process is not interrupted, we can generate answers with a delay of 450–800 msec, which falls within the acceptable threshold (see Figure 7).

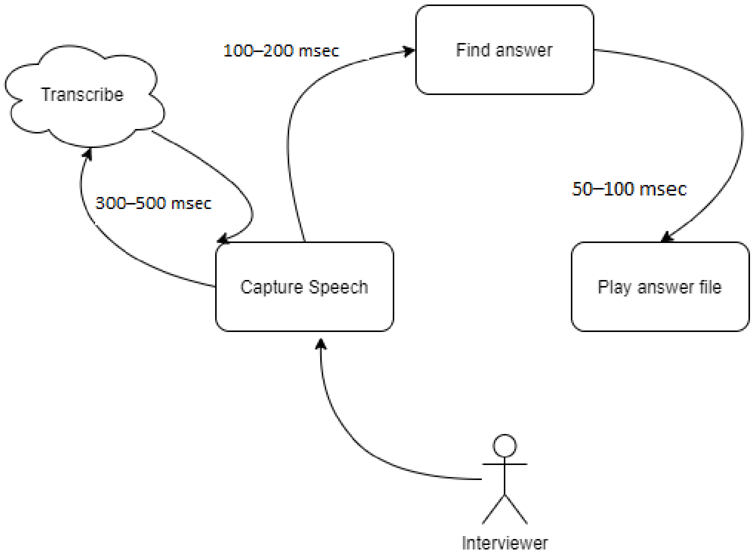


Figure 7. Average latency and delays across the modules involved in recording a question and returning an answer. Round trip time takes at most 700 msec with a stable internet connection of at least 1 Mbps.

The internet service we used at the event was a 4G LTE unlimited data service with download speeds of up to 10 Mbps and upload speed of 2 to 5 Mbps. We observed that, during peak times, the upload speed would fall between 50 and 75 Kbps which added up to 3000 msec of delay time to the systems, relying on *filler* answers more often. Rarely, the connection would become unstable. When this happened, the system would play chosen audio clips that inform the interviewer they should come back at a later time while a background process tries to reestablish a connection.

Considering these limitations, adding a live answer generation engine along with its text to speech module shouldn't add more than 5000 msec to the total system delay (and using filler answers). In the lab, we achieved this threshold by running the RNN based answering module on a Titan X with 3584 CUDA cores and 10 Gbps memory speed. However, the system used at the Biennale could not support the energy and heat dissipation demands of the GPU, and the graphics card used was busy rendering the 3D character.

3.3. Ceausescu

Nicolae Ceausescu was a Romanian dictator who was President of the State Council from 1967 to 1989, when he was overthrown. The EdgeAvatar system based on him, Ceausescu, shares a similar framework to Celan but uses prepared video answers in order to talk with the interviewer (i.e., users). An artistic decision was made to make Ceausescu seem as a static painting, with only the face being distorted to simulate talking. He would go on speeches and sometimes allowed to be interrupted for questions by users. In order to achieve this, instead of making a virtual rendering of the character, we interlaced videos that either played a speech or an interaction with the interviewer. For this virtual being, the question answering module chooses the video that matches the answer. The particular workflow for Ceausescu is as follows:

- While idling run a placeholder video that shows a static painting of Ceausescu (Figure 4).
- At different intervals, play a video that shows the painting reciting one of Nicolae Ceausescu's speeches.
- When a new user is sensed using object detection, welcome them with one of a welcoming phrases. Then, invite them to ask a question.

- After listening to the question, play the video that is chosen by the answering module.
- Repeat the question–answer loop until a random conversation threshold is reached or a timeout is triggered (signaling that the interviewer left).
- Resume the interrupted placeholder video (speech or static image) and repeat the workflow.

There are five key differences between Celan and Ceausescu: (1) movement detection, (2) use of a state machine, (3) question matching , (4) hardware requirements, and (5) video processing vs. 3D rendering.

3.3.1. Movement Detection

The movement detection module was responsible for sensing when a new user showed up on the scene. In order to detect a user, we were fetching frames from a camera connected to the computer. Each frame was converted into a black and white image, blurred, and sent to a motion detection module. We used a Gaussian blur with a kernel of size 13 × 13; however, this size was handpicked based on the camera properties and the environment around the scene. The blur was needed to minimize random noise between each frame of the video stream. The motion detection module maintained a list of the last 25 frames (this was an arbitrary number, which could be adjusted based on machine performance and desired accuracy) which are averaged together and then compared with the new frame. The comparison was a differential operation. Afterwards, the result was cleaned by using a threshold operation which blacked out all noisy pixels. Then, the contour finding algorithm from OpenCV [26] was used. We iterated over each contour and, if the contour size was greater than some arbitrary number, we assumed that there was a new user in the scene. If we detected a new user, the event was sent into a communication channel with a state machine. This approach allowed us to detect big changes in the scene without running any resource consuming machine learning model.

3.3.2. State Machine

The state machine module was responsible for maintaining the workflow logic of Ceausescu. This was needed because, unlike Celan, the visual and audio components were intertwined (the mouth animation and the audio were generated together), and seamless transition between videos was needed for a natural user experience. The state machine module used a graph to track the valid states that it could transition into. Subsequently, it used the graph-based states to determine the actions that it could and should take. Hence, each node in the graph consisted of the following properties:

- `video`—list of possible videos for this state
- `action`—function to run when we enter the state
- `on_event`—a dictionary of events and corresponding new state
- `blocked`—a boolean flag responsible for ignoring incoming events until the running video ends
- `kill_video`—a boolean flag responsible for killing a video if a previous video had a blocked flag set to false

This structure allowed us to easily adjust the logic of the system for different story lines. When we enter a new state, we first run an action associated with this state. It may include listening to the microphone, saving the video position, playing the next video, among other actions. After that, if there is more than one valid answer/video, we randomly get a video from the list and play it until we receive an event from the on_event list. Some of the events might include movement, end of video, and timeouts.

3.3.3. Question Matching

When the system falls into a question/answer loop, we use our question matching module to find a proper answer for a given question. We created a database of 30 questions and corresponding answers. Each question had a number of different formulations and different corresponding answers. There were 252 questions in the database. We used a term frequency based matching system, very similar to Celan's, but the output was the index of the video that contained the answer instead of a textual answer. Following the Speech to Text translation, question matching took on average 70 msec.

Since we had a finite number of answers, if there was a matching answer to the interviewer's query (determined by a confidence threshold), we selected a video from a folder containing responses deflecting the question. After answering a few questions or staying quiet for too long, a video of Ceausescu would let the interviewer and spectators know that he was done answering questions for the time being, and then he began to deliver a speech, ignoring further questioning until a time threshold was reached. During this speech period, both the camera and the microphone functions were disabled. This allowed the system to save energy since costly queries were not being processed.

3.3.4. Hardware

Ceausescu shares the same STT module as Celan. The hardware differs because Ceausescu doesn't require as much computing power since it has a low number of answers available and is not rendering a virtual object. The hardware description for Ceausescu is as follows:

- Casing: Neousys Nuvo-5306RT
- Processor: Intel SkyLake i7-6700
- Memory (RAM): 4 GB DDR3
- Storage: 500 GB SSD Hard Drive
- Graphics card: Nvidia GTX 1050 Ti 4 GB GDDR5

3.3.5. Video

The Celan system uses live 3D rendering and uses lip-syncing simulation on the rendered character. For Ceaucescu, the system instead relies on seamless transitions between videos. This was best suited for the art style and intention behind the system's design. To make this possible, the videos share a common resting position and the timings had to be adjusted by hand to match the targeted conversation delay. This was achieved through mpv [27], a low profile, open source media player. Once all videos were individually edited and adjusted to follow the same format and testing transition timings, we let the question answering module to determine which video should be played that would be queued into the state machine.

3.4. Web-Based EdgeAvatar

A third system was developed as an alternate version of Celan that could run as a web application. This is to be a portable version of EdgeAvatar since the physical nature of the older systems made it relatively costly and difficult to transport, requiring a relatively high amount of human labor and set up time. Another problem that this Web-based application solves is the need for an Internet connection. After being installed, the application can use an offline STT module with acceptable accuracy. This allows for EdgeAvatar to not be dependent on Cloud services during runtime. Therefore, with a web-based version of EdgeAvatar, the difficulties of deploying and distributing are traded for a more resource contained execution environment.

To eliminate the dependency on an Internet connection, the first version of Web Celan used the experimental Web Speech API [28] (at the time only fully supported by the Google Chrome browser) for both speech recognition and speech synthesis. The API is relatively lightweight, making it well suited for mobile devices. The additional benefits of using web APIs are the faster development time and that fewer bytes need to be transferred to the client. However, one has to talk in clear, American English with minimal background noise to obtain decent accuracy, which isn't a drawback for the original Celan and Ceausescu. Moreover, the synthesized speech sounds unnatural and is dependent on the available synthetic voices on the user's operating system, restricting the artistic freedom.

We used Tensorflow.js [29] in order to run machine learning models directly in the browser. Custom or open source speech recognition or speech synthesis machine learning models can be used to obtain higher accuracy than the Web Speech API. However, higher accuracy comes with the drawback of being more compute intensive and significantly increasing the size of the application. The necessary compute power to run these models could limit the devices that can run the web-based version.

The original Celan puppet was rendered using the Unity game engine [30]. Besides native platforms and game consoles, WebGL [31] is also one of Unity's compilation targets which means that the Celan puppet compiled for EdgeAvatar on a Windows 10 operating system could be reused for the web version. WebGL, however, has a limited set of supported features compared to other compilation targets. The High Definition Render Pipeline, for instance, could not be compiled to WebGL and therefore had to be replaced with the default Unity pipeline. The lip-syncing component made use of the microphone input, which is currently not possible when compiling to WebGL. Instead, by linking the amplitude of the synthesized answer to the position of the mouth, the virtual mouth can be controlled from within the web application.

As with any system, during the design of Celan, various trade-offs had to be made. By making an Edge Computing system, there was an inherent exclusivity to experiencing the artwork similar to how traditional art needs to be visited to be experienced. In contrast, a web application would allow for a much larger audience to experience the artwork on-demand, making it less exclusive but more fitting to contemporary culture. Because we controlled the hardware of the Edge Computing systems, they could be made as powerful as needed to obtain high accuracy on the modules. A web application, on the other hand, can be opened on a wide variety of devices outside of our control. To ensure that all users have at least a satisfying experience, the system modules need to be less resource-demanding resulting in lower accuracy, though this limitation might not be true in the near future. The technical constraints stemming from both system design approaches give the resulting two artworks a distinctive experience.

4. Discussion

Deploying the systems in the Biennale allowed us to witness users performing different types of interactions with our virtual beings. These interactions could be broadly separated into two classes. First, those who were familiar with the represented character (Celan or Ceausescu) were mostly interested in questions regarding modern topics, opinions, and philosophical problems. Second, the people who were not familiar would ask factual information about the characters, oftentimes using follow up questions to learn more about them. This is where a *zone of comfort* could come into play. The answers were designed so that the applications could stir the conversation back to topics that were known by the question matching system if a question triggered a non-factual response. We also implemented a conversation random threshold akin to PARRY's [32] anger variable to terminate conversations. The purpose of this was two-fold, to prevent repetitive answers and to improve user experience when in a public setting (allowing for another participant to interact).

The reliability of the Internet connection proved to be a limiting factor. The connectivity occasionally dropped, and it could take up to several seconds for the online STT and TTS modules to respond, which sometimes made the system feel slow and more robotic. To circumvent this, the systems would go into random speech while waiting for the service to be restored. An alternative version allowed for the use of offline STT, but the accuracy was less than optimal. In order to prevent any connectivity issues during run-time, it would be beneficial if the entire system was running locally with a state-of-the-art STT model running on a separate system. This would eliminate the need for an Internet connection at run-time and would make the cycle time of each module more predictable. Dedicated edge devices can be introduced to combat this problem. There can be an Edge Computing device with a microphone that transforms an audio stream into a text stream and pipes it to the rest of the system. A similar edge device using a speaker instead of a microphone for STT could receive a text stream from the answer generation module and transform that into an audio stream which is then played on the connected speaker.

Like any system, EdgeAvatar had to make various trade-offs in its design. Leveraging these trade-offs is an avenue of future work which can result in drastically different artistic experiences. The decision to make EdgeAvatar a custom Edge Computing system made its deployment and distribution relatively difficult but allowed for a rich and exclusive experience. In contrast, if the system were to be available as a web application it would allow for a much larger audience to experience the art which would make it less exclusive but potentially less personal as well. There are also additional technical trade-offs that would change the artistic experience of the system. The most notable is the trade-off between availability and accuracy/speed. By controlling the hardware of the Edge Computing system, the system can be more powerful and therefore capable of achieving higher accuracy. A widely available system like a web application is more resource constrained since web capable devices have very diverse levels of compute power, which is outside the control of the system.

Regarding system evaluation, most of our assessment had to be based on observation. We designed a version of the system that would evaluate the quality of the conversation based on how many interactions we could record from a single subject. However, due to European laws governing data collection, we decided it wasn't safe to implement it in real time. Part of our future work is to find a venue where we can inform the users that their interactions are being used to train the system, allowing us to implement this functionality.

4.1. Aesthetics

These being part of the artist's, Belu Fainaru, portfolio for the event, his input and vision for the systems played a key part in the visual and content aspects of the system. The characters, Celan and Ceausescu, were chosen after consulting with Belu, and there was a specific intent to communicate the peculiar personalities of the historical figures represented. In this sense, the answer didn't have to be completely accurate and could sometimes show the chatbot's unwillingness to give a straight answer. We witnessed that the audience would mostly react with amusement to this behavior. This aspect gave us more technical freedom on how we implemented the question answering module, since the answers didn't always have to be accurate, just loosely related to the questions.

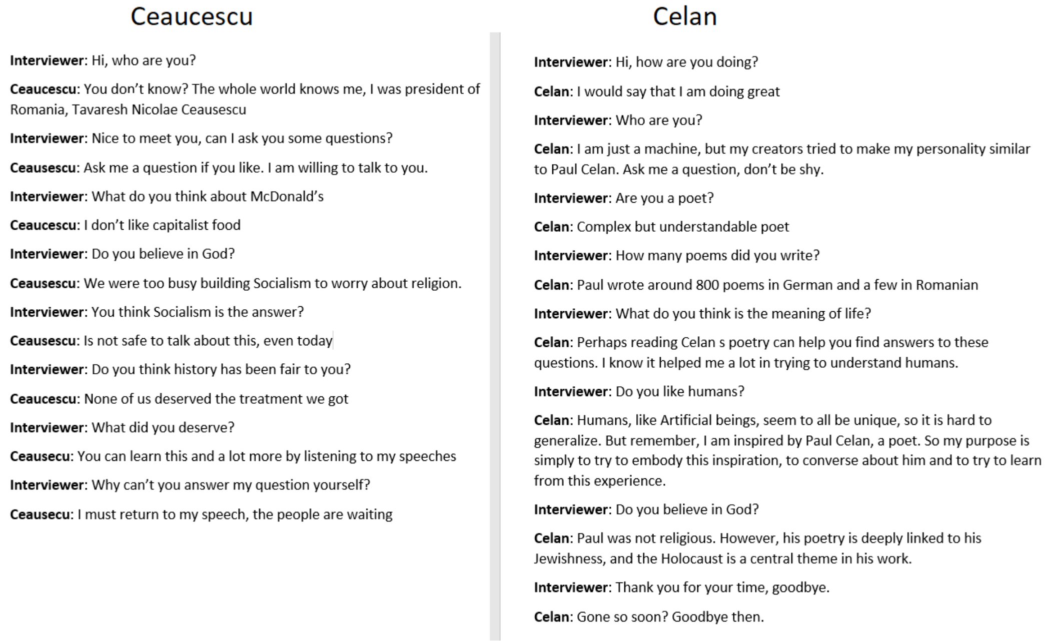
4.1.1. Design Constraints

In order for the system to work correctly, it needs a stable connection to the Internet, a microphone, a camera, a screen with a speaker, and sufficient processing power. However, any of these elements could easily be replaced by a different technology; for example, the display could be holographic or even a robotic face. The connection to the Internet is necessary in order to use STT for processing. In the off chance that the Internet connection breaks, the system's default response will be to play a speech telling the user that it does not understand the question posed and that it will stop listening. This is the main and most challenging bottleneck for EdgeAvatar. We mentioned that offloading speech processing tasks to a separate device could diminish this problem. However, in order to do this, a proper local network needs to be setup. This adds costs to the system and requires control of the exhibit space for hardware installation, which we didn't have in Venice.

4.1.2. The Design Process

EdgeAvatar provides a common framework for these systems; however the choices for how each module is going to work is part of a design process that considers the artistic inspiration. Initial blueprinting is about what the artist wants to communicate and how he or she sees the system working. Modules such as the STT and the question matching system are driven by the hardware constraints. However, visual and audio output are mainly artistic.

Figure 8 has a short excerpt of a conversation had with each of the two systems. They show how the experience for the interviewer is completely different depending on which application they are talking to. In the case of Celan, the intention of the system was to be respectful of the poet. Therefore, we intentionally chose creating a virtual avatar that would look similar enough to the poet to communicate inspiration, but not enough to seem like a parody. In fact, many of the factual answers made a point to indicate that this was not actually the real Paul Celan but a character inspired by him. This also played into the choice of synthetic speech over natural that would let the audience know that this is artificial and should in no way be considered an accurate imitation of Paul Celan. In some answers, the avatar introduces itself as an apprentice of the real Paul Celan.



Ceaucescu

Interviewer: Hi, who are you?

Ceaucescu: You don't know? The whole world knows me, I was president of Romania, Tavaresh Nicolae Ceausescu

Interviewer: Nice to meet you, can I ask you some questions?

Ceausescu: Ask me a question if you like. I am willing to talk to you.

Interviewer: What do you think about McDonald's

Ceaucescu: I don't like capitalist food

Interviewer: Do you believe in God?

Ceausescu: We were too busy building Socialism to worry about religion.

Interviewer: You think Socialism is the answer?

Ceausescu: Is not safe to talk about this, even today

Interviewer: Do you think history has been fair to you?

Ceaucescu: None of us deserved the treatment we got

Interviewer: What did you deserve?

Ceausecu: You can learn this and a lot more by listening to my speeches

Interviewer: Why can't you answer my question yourself?

Ceausecu: I must return to my speech, the people are waiting

Celan

Interviewer: Hi, how are you doing?

Celan: I would say that I am doing great

Interviewer: Who are you?

Celan: I am just a machine, but my creators tried to make my personality similar to Paul Celan. Ask me a question, don't be shy.

Interviewer: Are you a poet?

Celan: Complex but understandable poet

Interviewer: How many poems did you write?

Celan: Paul wrote around 800 poems in German and a few in Romanian

Interviewer: What do you think is the meaning of life?

Celan: Perhaps reading Celan s poetry can help you find answers to these questions. I know it helped me a lot in trying to understand humans.

Interviewer: Do you like humans?

Celan: Humans, like Artificial beings, seem to all be unique, so it is hard to generalize. But remember, I am inspired by Paul Celan, a poet. So my purpose is simply to try to embody this inspiration, to converse about him and to try to learn from this experience.

Interviewer: Do you believe in God?

Celan: Paul was not religious. However, his poetry is deeply linked to his Jewishness, and the Holocaust is a central theme in his work.

Interviewer: Thank you for your time, goodbye.

Celan: Gone so soon? Goodbye then.

Figure 8. This is a transcription of a conversation excerpt with Ceausescu **(left)** and Celan (**right**) which shows how the interactions have a different tone, even though the questions are similar

On the other hand, with Ceausescu, the intent was to highlight the personality of Nicolae Ceaucescu. In fact, in some of the answers, the system explicitly admits that he is Nicolae Ceacescu. Therefore, in order to achieve a more natural sounding voice, we relied on a voice actor to generate the answers. This added specific challenges to the system, since we wanted it to look as life like as possible. The first major requirement in designing this system was a need to play consecutive videos in such a way that a viewer could not tell if there was more than one video that ever played. In order to achieve this, we used OpenCV in Python, which gave us fine grain control over the video frames and allowed us to quickly swap from one video to another video, regardless of video formats. However, to achieve seamless transitions, we had to separate the audio from the images. We used Python's own media player being careful to synchronize it with the video being played.

Our choices are not necessarily the best options for the modules, but they do represent EdgeAvatar's versatility. Other than optimizing each individual module, we can add more devices to augment the user experience. Alternatively, EdgeAvatar can be streamlined into a lite version for specific applications, as it is the case with the web application. This is, to our knowledge, the first time a common framework is suggested for building virtual beings that incorporates Edge Computing into its design by offloading STT and TTS tasks to a middle layer.

4.1.3. Interactivity

Technical and quantitative performance for EdgeAvatar is inherent of the specific technology chosen for each module. While we briefly discussed how the systems used for the Venice Biennale behaved and performed based on the infrastructure, a major evaluating factor for the effectiveness of the system is the quality of the interactions. As stated in this work, we didn't record individual interactions due to privacy concerns, but, through observation, we analyzed the way onlookers interacted with the systems and evaluated the interactions based on how many interchanges the interviewer had. Table 2 shows the number of observed interactions. Exhaustion refers to the moment when the system communicates that it doesn't want to continue the conversation; we consider a conversation successful if it achieves exhaustion. However, some conversations went beyond this point, where interviewers would stay to converse after the exhaustion timeout resets. As a matter of fact, there was one interaction we observed with Celan where the interviewer stayed for over 40 min, and we have knowledge that a young interviewer came to converse with Ceaucescu on several days while the system was in Venice. During the observation window, over 70% of the conversations achieved exhaustion. For the duration of the event, the transcription service did around 1470 h of audio transcription with each audio clip lasting 2 s on average. This translates to about 8 h of transcriptions per day, since the event is open for 8 to 9 h a day, and this time doesn't account for responses; this means that the systems were involved in conversations virtually all the time.

Table 2. Number of interactions and transcriptions per system.

Criteria	Celan	Ceausecu
Observed interactions	120	70
Minimum number of exchanges	3	4
Maximum number of exchanges	15	12
Average number of exchanges	7	5
Average exchanges before exhaustion	10	8
Percentage of exhausted conversations	77.5%	75.7%
Total transcription time (hours)	755	715

5. Conclusions

In this paper, we described EdgeAvatar, a system for the creation of virtual beings that provides a flexible framework to build applications that take artistic intentions as first order design constraints. Since artistic input is a key element when building a virtual being, many of the decisions are driven by what the artist wants to communicate. Moreover, we use the principles of Edge Computing to create an immersive experience and a completely modular design that allows for integration of innovative technologies that solve each of the problems addressed in the overall application workflow. While EdgeAvatar is not the first system proposed for immersive dialogue systems, it is, to our knowledge, the first one that prioritizes aesthetics and artistic input in the creation of applications.

In summary, the outcomes of this work are:

- A common framework for building virtual beings that interact in public spaces
- We analyze a leverage model for ensuring the quality of the conversation
- We evaluate our systems based on the quality of interactions, a majority (75%) of interactions fulfilled the intended purpose
- EdgeAvatar allows for flexibility regarding local or cloud based services, considering the time constraints for generating responses
- A streamlined version of EdgeAvatar is described as an alternative for web applications

While it remains to be explored, EdgeAvatar is not restricted to the chosen infrastructure for the Venice Biennale; it is designed to accommodate novel components that improve interactivity. For example, the virtual display can be substituted with robotic faces, and the sensing module can include other sensors that improve context awareness. These are just some of the options that we plan to explore.

6. Future Work

Edge Computing, Machine Learning, Computer Graphics, Engineering, and Psychology are several areas involved in the EdgeAvatar system and its implementations. For these areas, this section discusses future work.

Future work should look for ways to incorporate recent advances in Machine Learning in the system's modules. The size and necessary compute power of current deep learning models make it challenging to use them on an Edge Computing system. Future work on the brain module should give the avatar more general knowledge. Moreover, it should be made aware of with whom it is interacting. Humans, for example, tend to respond with simplistic answers when talking to a child.

Advances in Computer Graphics and Machine Learning should improve the face module, creating a more human-like puppet. However, one should be mindful of the uncanny valley [33] when improving the human likeness of the system. Humans perceive it as eerie when something is near human-like. Future work should focus on creating more realistic mouth movements based on the output of the voice module.

Lastly, future work should look into making both the hardware and software of the system more robust and durable. If one of the virtual beings were to become part of a long-running exhibition, it should function for at least five years requiring minimal maintenance.

A major bottleneck for the virtual beings used in Venice was the internet connection. However, with the increasing availability of 5G connectivity, and the advances in edge device computation, we can perform more tasks locally and speed up retrieval from the cloud when necessary.

Author Contributions: Conceptualization and manuscript, N.W.; methodology and implementation, N.W., F.Z., A.S., M.D., and M.H.; onsite deployment, N.W., M.D., A.N., and A.V.; validation, N.W., F.Z., A.S., M.D., and M.H.; writing—review and editing, T.G. and A.N.; supervision, T.G., A.N., and A.V.; funding acquisition, A.N. and A.V. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding. The development of the two systems presented at the Venice Biennale 2019 was funded by the Donald Bren School of Information and Computer Sciences.

Acknowledgments: We thank Belu-Simion Fainaru for his artistic input and for inviting us to deploy the systems at the Venice Biennale 2019. We also thank Cristian Alexandru Damian (aka Sandro) and Flaviu Cacoveanu (Plan-B) for their crucial support and dedication to ensure that we had everything we needed to make the systems work. We especially want to thank Grigore Arbore, Director of Istituto Romeno di Cultura e Ricerca Umanistica, Venice, Italy for his hosting of our installations and symposium at the Institute during the Venice Biennale.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Turing, A.M. Computing machinery and intelligence. In *Parsing the Turing Test*; Springer: Dordrecht, The Netherlands, 2009; pp. 23–65.
2. Jurafsky, D.; Martin, J. *Speech and Language Processing*; Pearson: London, UK, 2009.
3. Kerlyl, A.; Phil, H.; Bull, S. Bringing chatbots into education: Towards natural language negotiation of open learner models. In Proceedings of the International Conference on Innovative Techniques and Applications of Artificial Intelligence, London, UK, 1 December 2006; pp. 179–192.
4. Eric, P. A Guide to Virtual Beings and How They Impact Our World. Available online:https://techcrunch.com/2019/07/29/a-guide-to-virtual-beings/ (accessed on 1 January 2021).
5. Shi, W.; Cao, J.; Zhang, Q.; Li, Y.; Xu, L. Edge computing: Vision and challenges. *IEEE Internet Things J.* **2016**, *3*, 637–646. [CrossRef]
6. Hari, S. Experiential media systems. *ACM Trans. Multimed. Comput. Commun. Appl. (TOMM)* **2013**, *9*, 1–4.
7. Guy, H.; Kubat, R.; Breazeal, C. A hybrid control system for puppeteering a live robotic stage actor. In Proceedings of the RO-MAN 2008—The 17th IEEE International Symposium on Robot and Human Interactive Communication, Munich, Germany, 1–3 August 2008; pp. 354–359.
8. Ravi, B.; Kulkarni, K; Pandey, V.K.; Rawls, C.; Mechtley, B.; Jayasuriya, S.; Ziegler, C. ODO: Design of Multimodal Chatbot for an Experiential Media System. *Multimodal Technol. Interact.* **2020**, *4*, 68.

9. Stavros, V.; Malliaraki, E.; dal Falco, F.; Di Maggio, J.; Massimetti, M.; Nocentini, M.G.; Testa, A. Art-bots: Toward chat-based conversational experiences in museums. In Proceedings of the International Conference on Interactive Digital Storytelling, Los Angeles, CA, USA, 15 November 2016; Springer: Cham, Switzerland, 2016; pp. 433–437.
10. Kyungyong, C.; Park, R.C. Chatbot-based heathcare service with a knowledge base for cloud computing. *Clust. Comput.* **2019**, *22*, 1925–1937.
11. Vaios, K.; Foteas, A.; Papaioannou, A.; Kapari, M.; Sakkas, C.; Koumaras, H. 5G performance testing of mobile chatbot applications. In Proceedings of the 2018 IEEE 23rd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), Barcelona, Spain, 17–19 September 2018; pp. 1–6.
12. Tata, S.; Patel, J.M. Estimating the selectivity of tf-idf based cosine similarity predicates. *SIGMOD Rec.* **2007**, *36*, 7–12. [CrossRef]
13. Tovée, M.J. Neuronal Processing: How fast is the speed of thought? *Curr. Biol.* **1994**, *4*, 1125–1127. [CrossRef]
14. Lilla, M.; Bastiaansen, M.C.M.; Ruiter, J.P.D.; Levinson, S.C. Early anticipation lies behind the speed of response in conversation. *J. Cogn. Neurosci.* **2014**, *26*, 2530–2539.
15. Matthew, R.; Skantze, G.; Harte, N. Investigating speech features for continuous turn-taking prediction using lstms. *arXiv* **2018**, arXiv:1806.11461.
16. Angelika, M.; Hough, J.; Schlangen, D. Towards deep end-of-turn prediction for situated spoken dialogue systems. In Proceedings of the INTERSPEECH 2017, Stockholm, Sweden; 20 August 2017; pp. 1676–1680.
17. Simon, W. The Venice Biennale. *Burlingt. Mag.* **1976**, *118*, 723–727.
18. Shaman, S.S. Belu-Simion Fainaru. *Journal of Contemporary Art*, New York, NY, USA, 2 July 1995.
19. Mulazzani, M. *Guide to the Pavilions of the Venice Biennale Since 1887*; Electa: Milan, Italy, 2014.
20. Duroche, L.L. Paul Celan's Todesfuge: A New Interpretation. *MLN* **1967**, *82*, 472–477. [CrossRef]
21. Speech to Text—Converts Spoken Audio to Text for Intuitive Interaction. Available online: https://azure.microsoft.com/en-us/services/cognitive-services/speech-to-text/(accessed on 1 January 2021).
22. Andrej, K. "Char-RNN" Github Repository. Available online: https://github.com/karpathy/char-rnn (accessed on 1 January 2021).
23. Cristian, D.N.M.; Lee, L. Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogues. *arXiv* **2011**, arXiv:1106.3077.
24. Bryan, S. Literary freedom: Project gutenberg. *XRDS Crossroads ACM Mag. Stud.* **2003**, *10*, 3.
25. Amazon Polly. Available online: https://aws.amazon.com/polly/ (accessed on 1 January 2021).
26. OpenCV. Available online: https://staging.opencv.org/ (accessed on 1 January 2021).
27. MPV. Available online: https://mpv.io/(accessed on 1 January 2021).
28. Julius, A. *Web Speech API*; KTH Royal Institute of Technology: Stockholm, Sweden, 2013.
29. Smilkov, D.; Thorat, N.; Assogba, Y.; Yuan, A.; Kreeger, N.; Yu, P.; Zhang, K.; Cai, S.; Nielsen, E.; Soergel, D.; et al. Tensorflow. js: Machine learning for the web and beyond. *arXiv* **2019**, arXiv:1901.05350.
30. Unity Game Engine. Available online: https://unity.com/(accessed on 1 January 2021).
31. Tony, P. *WebGL: Up and Running*; O'Reilly Media, Inc.: Newton, MA, USA, 2012.
32. Mark, C.K.; Weber, S.; Hilf, F.D. Artificial paranoia. *Artif. Intell.* **1971**, *2*, 1–25.
33. Mori, M.; MacDorman, K.F.; Kageki, N. The Uncanny Valley [From the Field]. *IEEE Robot. Autom. Mag.* **2012**, *19*, 98–100. [CrossRef]