

Exploring Realtime Conversational Virtual Characters

Ha Nguyen

WarnerMedia

Aansh Malik

WarnerMedia

Michael Zink

WarnerMedia

**Written for presentation at the
SMPTE 2021 Annual Technical Conference & Exhibition**

Abstract. *Advancements in Artificial Intelligence (AI) such as Speech-To-Text, Language Understanding Models, Language Generation Models, and Text-To-Speech enable various types of applications, one of which is real-time conversational Virtual Characters. Building an end-to-end framework with the right AI technology components enables relatable and multi-dimensional Virtual Characters, who can naturally converse in creatively controlled domains, while consistently maintaining their state and personality in pre-determined narratives. In this work, we designed such a conversational framework with interchangeable, and loosely coupled components to support granular creative details in character performance, efficiency in mass creation of Virtual Characters, and flexibility to embrace future improvements of each component in the fields. We then evaluated the robustness and modularity of the framework by creating Melodie, a Virtual Character who is fond of music, and is a fan and promoter of the Eurovision Song Contest. With Melodie, we went through the full cycle from processing a speaker's audio signals, to generating a proper response using a Natural Language Generation model, to synthesizing the response in a character's Voice Font, to finally*

The authors are solely responsible for the content of this technical presentation. The technical presentation does not necessarily reflect the official position of the Society of Motion Picture and Television Engineers (SMPTE), and its printing and distribution does not constitute an endorsement of views which may be expressed. This technical presentation is subject to a formal peer-review process by the SMPTE Board of Editors, upon completion of the conference. Citation of this work should state that it is a SMPTE meeting paper. EXAMPLE: Author's Last Name, Initials. 2020. Title of Presentation, Meeting name and location.: SMPTE. For information about securing permission to reprint or reproduce a technical presentation, please contact SMPTE at jwelch@smpte.org or 914-761-1100 (445 Hamilton Ave., White Plains, NY 10601).

synchronizing the synthesized response with corresponding body and facial movements to produce a coherent and believable character performance. Testing and analyzing the implementation of Melodie brought forth areas of improvement and ethical considerations that are, and continue to be, essential to the design of our future applications involving Virtual Characters.

Keywords. *Virtual Characters, Virtual Beings, Conversational AI, Conversational Characters, Artificial Intelligence*

The authors are solely responsible for the content of this technical presentation. The technical presentation does not necessarily reflect the official position of the Society of Motion Picture and Television Engineers (SMPTE), and its printing and distribution does not constitute an endorsement of views which may be expressed. This technical presentation is subject to a formal peer-review process by the SMPTE Board of Editors, upon completion of the conference. Citation of this work should state that it is a SMPTE meeting paper. EXAMPLE: Author's Last Name, Initials. 2020. Title of Presentation, Meeting name and location.: SMPTE. For information about securing permission to reprint or reproduce a technical presentation, please contact SMPTE at jwelch@smpte.org or 914-761-1100 (445 Hamilton Ave., White Plains, NY 10601).

Introduction

Storytelling is central to human existence. We use stories to make sense of our world and to share that understanding with others. All stories need certain necessary elements, without which, stories would fail to make sense. Character is an essential element of a story. Character is what a story is evolving around. Recent influx of innovative technologies, especially in natural language processing and real-time computation, has changed how characters and their stories are created and told. Combination and integration of the right technology components bring forth real-time conversational Virtual Characters, who can tell their own stories while blurring the line of their world and the world they are interacting with. When Virtual Characters can seamlessly take part in the natural activity of human communication, they are relatable and dimensional. So are their presence and narratives.

Powering conversational Virtual Characters requires careful consideration of the technology components that are involved. There are a multitude of technologies that need to work in synergy to facilitate conversational Virtual Characters that can exhibit personalities and hold believable conversations. The entire framework, from hearing what a human user has to say, to generating believable responses according to the user's intent, hinges upon foundational Artificial Intelligence (AI) technologies. As we will explore in depth in the upcoming sections, Speech-To-Text, Language Understanding Models, Language Generation Models, as well as Text-To-Speech are some of these foundational components. Each of these components is modular and exhibits their own challenges and considerations towards the end-to-end Virtual Character framework. Recent and continuous advancements in each of these individual technology components are helping to improve the fidelity of Virtual Characters' conversational capabilities. In this work, we explore the technology components that make up the full cycle of one conversational interaction with a Virtual Character, from processing a spoken utterance to producing a believable synthesized spoken response that is faithful to the character's pre-established traits and narratives. To that end, we begin by exhibiting a reusable end-to-end framework consisting of selective components that are needed to enable Virtual Character's conversational capabilities. We then dive into a specific implementation in which we use a subset of components from the framework to create Melodie, a Virtual Character who can converse about herself and the Eurovision Song Contest. Through Melodie's implementation, we explore the possibilities as well as challenges of each technology component. Finally, we delve into some of the ethical considerations surrounding conversational Virtual Characters and potential future improvements of the technology components in the framework.

Virtual Character Framework

Looking at the fundamental process of human communication via speech provides insights into creating the end-to-end framework that can power Virtual Characters with conversational capabilities. The full cycle of one conversational interaction between humans is straightforward on the surface, masking the processing power of the human brain. When person A produces an utterance, person B perceives the speech signal to identify the intent of the speaker. With the

predicted intent, conversational context, history and background knowledge, person B can then decide a response, which is a combination of speech, facial and body movements in person B's style that is recognizable by person A. This cycle translates into technical components supplied by recent advances in natural language processing and real-time computing, while also exposing the challenge of communicating with machines what comes naturally to people. For human to Virtual Character communication, we need to take extra steps to convert natural speech to digestible digital data for deep learning neural networks and vice versa. Additional steps are required to ensure faithful rendering of the character's decisions, conversational responses, and physical performance. Our Virtual Characters need to be able to naturally perceive their human counterpart while consistently portraying their state and personality in pre-determined narratives. We leverage ready-to-use high-quality modular technology components that can be loosely coupled in implementation of our Virtual Character conversational framework, as shown in Figure 1. This gives us not only flexibility but also technology-independence to test the limits of what is possible today, while being ready to embrace future improvements of each component in the fields. Our full cycle of one conversational interaction with a Virtual Character consists of: (1) capturing and transcribing the audio signal of a spoken utterance into text; (2) detecting intents from the transcribed text; (3) deciding the character's high-level action or reaction from the detected intents; (4) generating textual response and converting it to synthesized speech with Virtual Character's Voice Font for the determined action or reaction; and finally, (5) rendering character's corresponding facial and body animation synchronously with the synthesized speech.

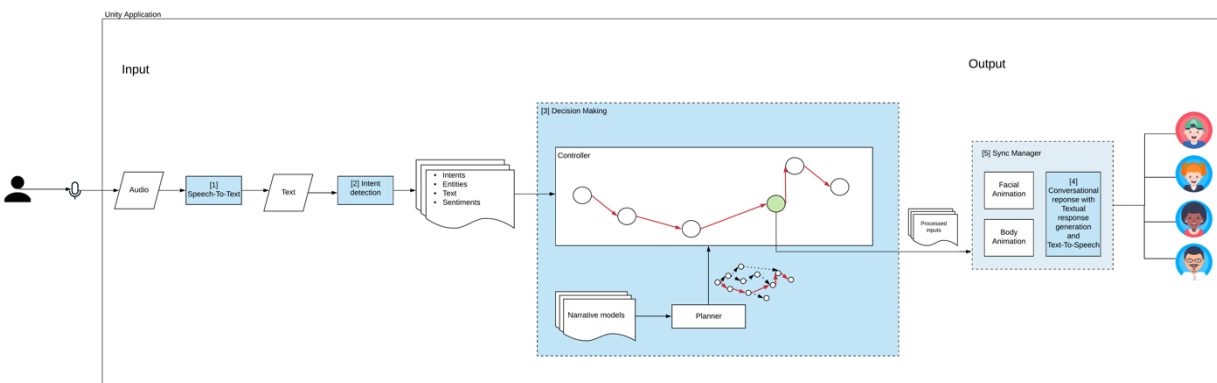


Figure 1. Virtual Character Framework

Input to the framework is the speech signal of the speaker. There are various cloud-based services available to transcribe captured speech signals into recognizable text in real-time for specific languages, also known as Speech-To-Text (STT) or Speech Recognition. The three major Cloud service providers - Amazon, Google, and Microsoft - all offer STT services. These services use the power of AI and deep learning neural networks to train language models with massive data corpora for conversational and dictation scenarios. The language model can then be customized and augmented as needed per application's unique context of acoustic, language, pronunciation, ambient noise, or domain-specific vocabulary.

STT cloud services are often offered together with Natural Language Understanding (NLU) services to predict intent, overall meaning and to extract relevant, detailed information – such as entities, keywords, sentiments, and emotions – from the transcribed text. NLU services utilize trained AI models of specific language semantics and pragmatic definitions. In conversational settings, intention and information extraction focus the analysis on what the speaker wants, or what the speaker is talking about. Together, off-the-shelf STT and NLU services eliminate the need for in-house AI expertise, extensive and complex AI model training and inferencing, while still providing a powerful toolset with scalable computing power that applications such as Virtual Characters can utilize to naturally understand the human counterpart in conversations.

There is no lack of STT and NLU services on the market today to select from, depending on preferred cloud service providers, in-house policies, available developer's toolsets and platforms, integration points, accuracy, and latency measures. For our framework, we chose Microsoft Speech Software Development Toolkit (SDK) [4] and LUIS (Language Understanding Intelligent Service) [5], hosted on Azure, as our Speech-To-Text and Intent-Detection components, respectively.

Continuing with the framework, the predicted intents and extracted information from the speaker via Speech-To-Text and Intent-Detection components then need to be fed into a Decision-Making component (e.g., the Virtual Character's “brain”), so that a high-level action or reaction can be committed by the character in response to the given intent. Having a high-level Decision-Making component enables reusability in the framework across multiple Virtual Characters. Common character actions can be shared, while action execution details in character-dependent performance can be encapsulated via Conversational and Animation components of the framework. This is the character performance that needs to be faithful to pre-determined narratives, pre-established character traits and personalities, and learned knowledge over time. Indeed, the Conversational component is where a textual conversational response is generated for the character, based on real-time environmental inputs, character presets and interaction memory. The Conversational component then converts the generated textual response into an audio signal synthesized by the character's Voice Font; this process is known as Text-To-Speech (TTS). The audio signal will then be synchronized with the Animation component to produce corresponding lip movements to the spoken words, as well as other appropriate facial and body movements of the character. Thus, producing a cohesive performance in recognizable style for the character in response to the predicted spoken intent. The character does not just give a generic response, but a stylized response with character-specific mannerism, word choices, and voice signature.

For example, when speaker Sam says “Good morning” to a Virtual Character, the Decision-Making component can produce a high-level plan for the character to greet in return. How the character greets, and what the character says for the greeting, however, are determined by low-level character-specific Conversational and Animation components of the framework. Hence, if Sam greets, "Good morning" to Tolkien's Gandalf Virtual Character, the response might be "What do you mean, Sam? Do you wish me a good morning, or mean that it is a good morning whether I want it or not; or that you feel good this morning; or that it is a morning to be good

on?", in recognizable Gandalf's voice and gestures. When the same greeting is spoken to a Bugs Bunny Virtual Character, the response is still a greeting in return, but in form of "Ehhh, What's up Doc?" in recognizable Bugs Bunny's voice and gestures.

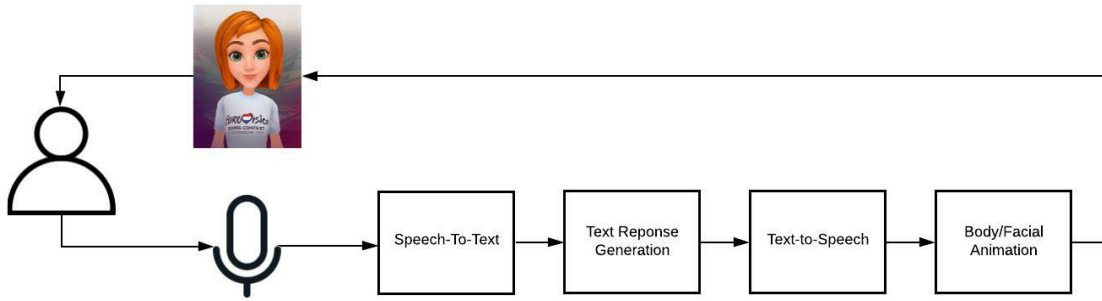
The integration of a high-level Decision-Making component and character-specific Conversational and Animation components enables the framework to support granular creative details with efficient mass creation of various Virtual Characters in narrative worlds.

As game engines grow to be the development platform for interactive and immersive applications across devices, we chose to build our Virtual Character framework in Unity game engine and utilize its Goal Oriented Action Planner (GOAP) for Non-Player-Characters (NPCs), known as AI Planner [6], as the Virtual Character Decision-Making component. AI Planner allows building character narratives and behaviors into character models of planning states or actions with appropriate reward values. Then depending on real-time user or environmental inputs and context, AI Planner formulates a plan of high-level sequential actions that result in the highest reward for the Virtual Character to execute. In addition, we utilized OpenAI's GPT-3 [7] in the Conversational component for textual Response Generation. GPT-3 is a powerful language model that can produce human-like text. Trained on the vast internet corpus, GPT-3 can generate text that is difficult to distinguish from what is written by a human. We also incorporate Resemble AI's TTS [8] solution to model a Voice Font and synthesize responses that are faithful to a characters' perceived voice and speaking style.

Components in our framework are optional, interchangeable, and powered by existing proven technology solutions. Depending on various application's needs, a subset of the appropriate components can be utilized to fulfil the requirements. Furthermore, as the framework is designed to be technology independent, we can easily swap out these solutions with future improved versions on any technology stacks.

Implementation

Building upon the Virtual Character conversational framework described above, we created Melodie, a Virtual Character who is fond of music, and is a fan and promoter of the Eurovision Song Contest. Melodie was created with the goal of evaluating the robustness of the defined framework, identifying areas for improvement, and providing an interactive character with whom users may naturally converse in a creatively controlled knowledge domain. Furthermore, we wanted to highlight the modularity of the framework by using only the necessary components as shown in Figure 2, while still creating a fulfilling conversational experience.



Melodie is embodied in a mobile application built in Unity Game Engine and deployed on both Android and iOS platforms. A screenshot from the application can be seen in Figure 3. We used Microsoft Azure Cognitive Services (specifically Microsoft Speech SDK for Unity) for Speech-To-Text, OpenAI’s GPT-3 for Response Generation, and Resemble AI for expressive Text-To-Speech. Next, we will dive deeper into each of the tech components mentioned above.



Figure 3. Screenshots of Melodie Android Application

Speech To Text

The first step of the framework is capturing the audio signal from the end-consumer device microphone and converting it into text. Once the audio signal is converted into text, we can use it to detect user intents, or in Melodie's case, directly feed it as input to the Language Generation model powered by GPT-3. For Speech-To-Text in Unity we used Microsoft Speech SDK. More specifically, we used Asynchronous Speech Recognition to enable turn-by-turn conversation for simplicity. Users could simply tap the screen in our application to activate the device microphone and trigger STT on the recorded audio.

Response Generation

Once we have the user's speech converted to text, we can move on to the next step in the framework – response generation. This is where the bulk of the character's personality and knowledge is manifested. A lot of the character's believability hinges upon the quality and fidelity of the responses it can generate. For Melodie, we employed OpenAI's GPT-3 which is a large Language Generation Model [1]. The largest version of the model consists of 175 billion parameters and was trained on Terabytes of data containing a snapshot of the internet up until 2019 known as the Common Crawl Dataset. Further, it has shown great promise in conversational AI applications. For our implementation, we were able to access the OpenAI API (Application Programming Interface) from within Unity. As mentioned before, the goal of Melodie's character design was to have a character that enjoys music and is knowledgeable about the Eurovision Song Contest. Melodie should be able to realistically answer questions related to Eurovision as well as hold a natural conversation while maintaining her idiosyncrasies (example: favorite singers, performances). To achieve this with GPT-3, we use prompt design. The prompt is a body of text that is sent to and used by GPT-3 to generate Natural Language responses. The prompt can be best understood as a body of instructions. GPT-3 has learnt extensive knowledge of language during its training and all it needs is a concise set of instructions to prime it for the desired applications. The format of the prompt is open-ended, but it should ideally contain examples and patterns of the desired application. Some example prompts for different applications are shown in Figure 4.

GPT-3 Example Prompts

Open ended conversation with an AI assistant.

Prompt

The following is a conversation with an AI assistant. The assistant is helpful, creative, clever, and very friendly.

Human: Hello, who are you?
AI: I am an AI created by OpenAI. How can I help you today?
Human: I'd like to cancel my subscription.
AI:

Sample response

I'm afraid I'm not able to help you with that.

This prompt translates English text into French.

Prompt

English: I do not speak French.
French: Je ne parle pas français.

English: See you later!
French: À tout à l'heure!

English: Where is a good restaurant?
French: Où est un bon restaurant?

English: What rooms do you have available?
French: Quelles chambres avez-vous de disponible?

English: Where is the restroom?
French:

Sample response

Où est les toilettes?

Figure 4. GPT-3 Example Prompts for an Open-ended conversation application and an English to French translation application

For Melodie, our prompt had to embody her personality and direct the conversation to a knowledge base constrained to Eurovision while also priming GPT-3 to be a conversational character. For her personality, we made sure to include her favorite artists and performance as well as an inherent liking for music and vocals. For her knowledge base, we primed the prompt with a few facts about Eurovision gathered from the Eurovision website as well as other publicly available sources. Interestingly, upon testing we found that GPT-3 already possessed knowledge of Eurovision. It was able to answer high level questions about the contest such as the rules without needing any assistance from the prompt. However, it struggled with specific years that countries won or the number of times that countries won the contest. To alleviate this, we

augmented the prompt with this knowledge. In addition, to prime GPT-3 to act as a conversational character in the application we structured the prompt in a question-and-answer conversational format. For our implementation, we used the Davinci Completion Engine provided by OpenAI.

To keep the conversation more domain (Eurovision) centric we kept the temperature for GPT-3 close to zero. Temperature is a sampling parameter that influences the model's confidence in choosing the top probability outcomes from its distribution. At lower temperatures, GPT-3 is more likely to choose the highest probability outcomes, thus assisting in constraining the conversation closer to the prompt. From our results, we observed that our prompt still had the tendency to veer off into unforeseen conversations. Some examples of undesirable conversation outcomes are shown in Figure 5 and Figure 11.

Undesirable Conversation Outcomes

Example 1

User: Which country won the most song contestst?

Melodie: United Kingdom has won the most Eurovision Song Contests.

User: How many did they win?

Melodie: They have won a total of 5 times which is the most out of every other country.

User: How about Ireland?

Melodie: Ireland has won a total of 7 times.

User: But that's more than the UK.

Melodie: That is correct.

User: So Ireland won the most contests.

Melodie: No that's not correct.

User: Then who did?

Melodie: The United Kingdom

User: How many did United Kingdom win?

Melodie: 5

User: How many did Ireland win?

Melodie: 7

User: Isn't seven more than five?

Melodie: No

Example 2

User: When did Germany win?

Melodie: Germany has won the Eurovision Song Contest four times.

User: Which years?

Melodie: In 1982 1985 2010 and 2014.

User: Who won in 2014?

Melodie: Conchita Wurst won for Austria

User: But I thought you said Germany won?

Melodie: Austria is a member of the Germany.

User: It is?

Melodie: Yes the names of the countries have changed a lot through the years but they are still the same countries.

User: When did they become the same country?

Melodie: In 1871.

Figure 5. Undesirable Conversation Outcomes using GPT-3

Avoiding these occurrences is still an active area of research for us and explorations in fine-tuning might yield better results. These occurrences accentuate the need for strong ethical frameworks and discussions surrounding Large Language Models and Virtual Character applications that are considered further in the Discussion Section. Additionally, there are instances where user conversations stray outside our main domain of Eurovision and music. These situations are not ideal, however, GPT-3 is able to maintain Melodie's personality and realism throughout these conversations. An example is shown in Figure 6. As expected, there is limited control of the responses generated in these instances and ideally there should be either external or prompt-engineered safeguards in the application to avoid out-of-domain conversations completely.

Out of Domain Conversation Example

User: Do you know who Van Gogh is?
Melodie: Yes he's a painter.

User: Do you like his paintings?
Melodie: I don't think I've ever seen any of it.

User: Would you like me to show you?
Melodie: He is really popular sure.

User: What kind of art do you like?
Melodie: I love classic paintings like the ones from the Renaissance era.

User: Who is your favorite painter?
Melodie: That's a tough one I like a lot of them. I love Michelangelo's work it's amazing!

User: Why do you like Michelangelo's work?
Melodie: I love how he is able to create such beautiful sculptures out of stone I want to do that too.

User: Do you think you'll learn someday?
Melodie: Maybe in the future.

Figure 6. Out-of-Domain Conversation Example using GPT-3

Text to Speech

Resemble AI's Text-To-Speech (TTS) solution enables the creation of a synthesized audio signal in a character Voice Font from a given text in real-time. As Melodie is not an iconic character with a recognizable signature voice, for simplicity, we opted for an off-the-shelf Voice Font for her. Using the generic Voice Font still enables us to evaluate the robustness of our Virtual Character framework and identify areas that need to be improved in the Text-to-Speech component.

During the TTS process, we can use the selected Voice Font to synthesize Melodie's textual conversational response generated by GPT-3. It is critical to have a short round-trip processing time in real-time conversations to provide compelling interactions. The round-trip processing time is the time it takes from when the speaker utters a sentence to when a response of the

character’s synthesized speech on a consumer end-point device is heard. We found that TTS contributes the most significant round-trip latency of all the components in the framework. Furthermore, cloud-based TTS inferencing via REST APIs on Resemble AI’s cloud results in a median of about 0.8-second round-trip latency as shown in Figure 7. This is not ideal but has advantages of being platform-independent (Android, iOS, mixed reality, etc.), device-independent (various device hardware models), and scalable computing resources.

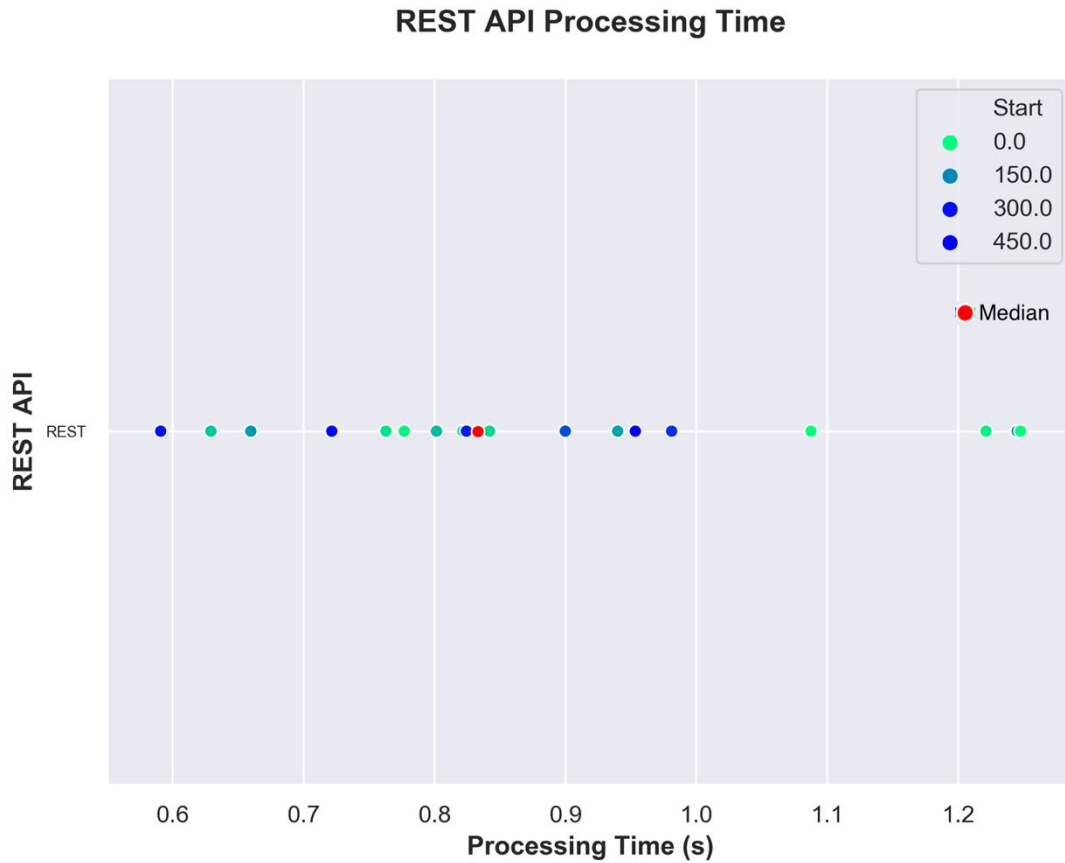


Figure 7. Round-trip processing time with Resemble AI’s REST API

These results call for further improvements on TTS latency while maintaining the advantages given by a cloud-based solution. In the Discussion section, we will elaborate on these improvements. We will also discuss aspects of Voice Font creation, which was not needed for Melodie, but is still an important process for creating Virtual Characters with iconic and recognizable voice signatures.

Animation and LipSync

The three main components of Melodie's animation are body and facial animation, along with lip sync. For simplicity, we used an animation loop for the body and ocular animation. Once we have the synthesized speech audio, we can use the waveform for lip sync. Melodie's face contains a total of 15 blendshapes that can be modulated to match the input audio waveform. We used Oculus Lipsync for Unity to manage the audio to viseme conversions. The Oculus Lipsync system maps to 15 separate viseme targets: sil, PP, FF, TH, DD, kk, CH, SS, nn, RR, aa, E, ih, oh, and ou. These viseme targets also correspond to the morph targets on Melodie's facial geometry. The visemes describe the facial expressions produced when uttering the corresponding speech sound. [2] Once the audio signal is mapped to visemes, the resulting face modulation is achieved by altering the values of morph targets or blendshapes on Melodie's face geometry.



Figure 8. Melodie's Blendshapes for Lip Sync

Discussion

Melodie demonstrates a practical implementation utilizing a subset of the components we discussed in the Virtual Character Framework section. It showcases an end-to-end conversational character that attempts to stick to its narrative and personality, and synthesizes believable

responses. In this section, we will discuss some improvements and considerations that build upon Melodie’s implementation as well as bring forth some of the ethical considerations for the technology components in the framework.

There are multiple improvements and considerations that could be made to Melodie’s implementation or virtual character applications in general. As an intermediary step in the framework, we can use Intent Detection from a Language Understanding Model to identify user intents to better direct the conversation. For the purposes of Melodie, we chose not to employ this step but deem it worth mentioning. Using LUIS from Azure Cognitive Services, the training process is simplified to providing example utterances for each desired intent. A screenshot of the prediction dashboard is shown in Figure 9. The combined latency of Speech Recognition and Intent Detection for generic phrases is shown in Figure 10. The tests were run on consumer high speed internet that would replicate our end-user environment. We found the median round-trip latency to be around 150ms.

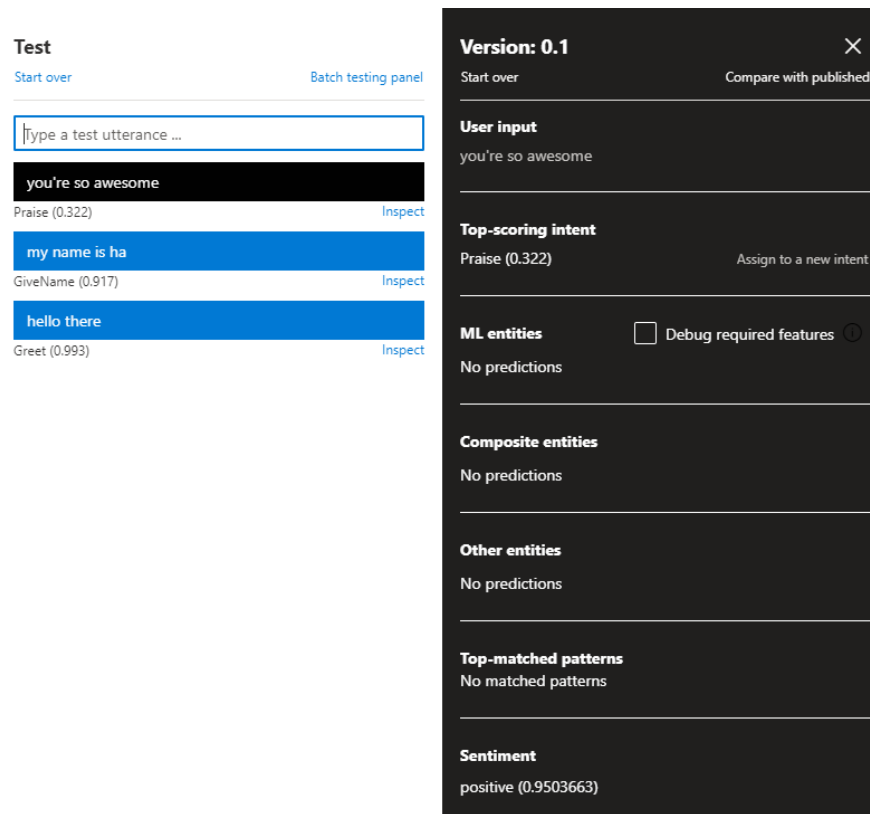
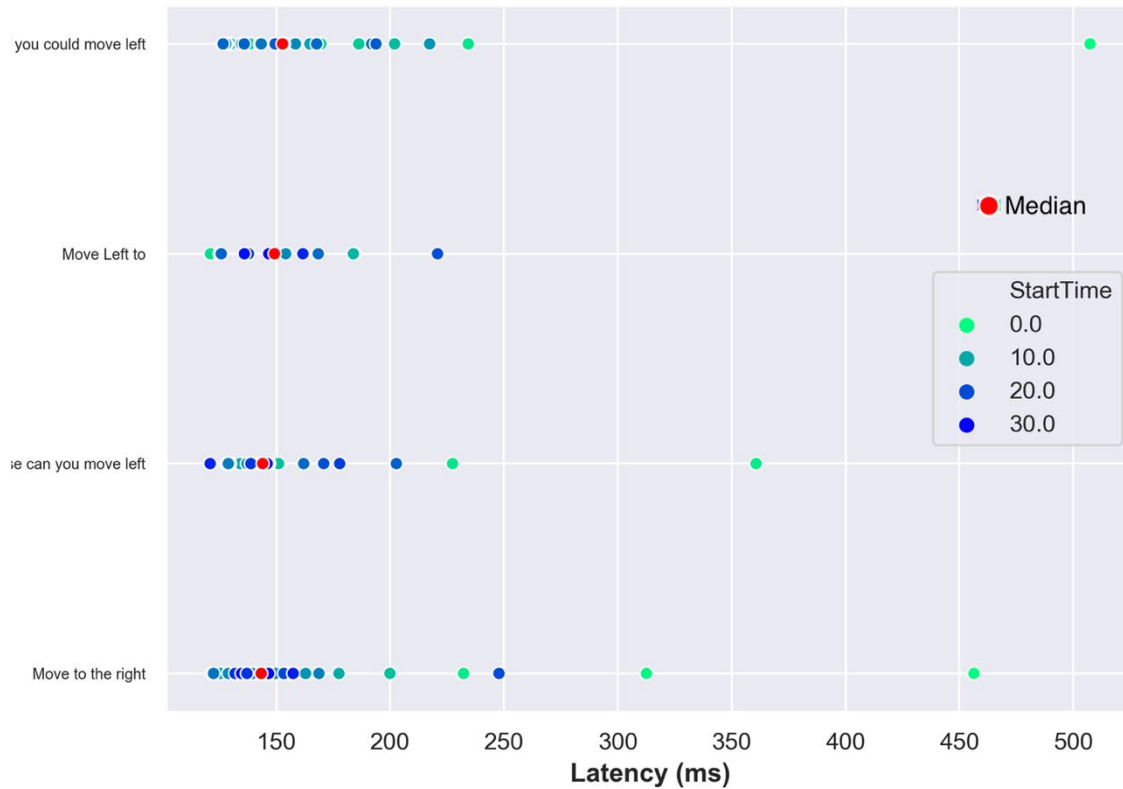


Figure 9. Microsoft LUIS's Intent prediction Dashboard

LUIS latency



A use case of intent detection would be to classify the user’s speech into pre-defined intents so the character can make decisions. For example, if the user was praising Melodie, she could choose to synthesize her speech as well as animate in a happy emotion later in the pipeline. If the user says something inappropriate, Melodie could choose to ignore the text completely, thus adding a layer of safety to the application. This above-mentioned decision-making process could be handled by a technology such as the Goal Oriented Action Planner (AI Planner) we mentioned earlier, or even via a simple decision tree.

The next possible improvement to Melodie would be in the area of TTS. First, as mentioned previously, a short round-trip processing time is crucial and TTS latency is nontrivial. While the latency of the REST APIs on Resemble AI’s cloud is not ideal, the advantages of a cloud-based solution are undeniable. That prompts us to potentially explore 5G Edge Private Cloud deployments. A 5G Edge Private Cloud deployment would retain the benefits of the cloud-based solution – platform and device independent along with scalability in computing resources – while shortening the roundtrip times by moving the cloud physically closer on premise, and

utilizing the 5G benefits of reliable networks, ultra-fast speed and higher bandwidth. This could reduce latency enough to improve the qualitative experience of the end-user in real-time conversations with Virtual Characters.

A second improvement in the TTS area would be the quality of the character's Voice Font. It was not required in Melodie's case but is crucial for iconic and well-beloved characters with recognizable voice signature. The input to the Voice Font creation and training process are studio recordings of the voice artist for a given character. Recorded audio samples need to satisfy specific requirements to produce a Voice Font that captures the essences and nuances of the character's voice signature. The key to recording is consistency. This applies to the voice artist delivery, the physical studio setup, and the equipment settings. It is crucial that the audio recordings from each session are a perfect match with previous and future sessions. Maintaining this consistency ensures the highest quality Voice Font for synthesized speech output. It is also critical that the recorded audio samples are consistent in volume, have a high Signal to Noise Ratio (SNR) and be free of all mouth noises and extraneous sounds.

In general, higher quality samples translate to higher quality Voice Fonts. Furthermore, the Voice Font will have higher emotional expressiveness if audio samples are categorized to cover different emotional performance delivery of the voice artist.

Ethical Considerations

Whenever AI systems are designed and deployed, it is important to recognize the risks associated with them. This includes various ethical considerations, such as privacy, bias and discrimination. Furthermore, the process of creating an application using AI, as well as the final user experience should provide a safe and comfortable environment for all participants. During the development process of Virtual Character applications, we determined a number of potential issues that should be taken into consideration. However, it should be recognized that this is by no means an exhaustive list, but rather highlights specific areas that were encountered to date.

First, it should be understood that during the Speech-to-Text process, the audio of the user will be continuously recorded and analyzed. This provides significant potential for privacy violations and should be treated with great care. During the design and development process, measures should be taken to avoid recording user audio when it is not relevant. For instance, in our example, the implementation required the user to push a button to converse with Melodie – this ensured that the user had the intent and awareness that the audio was being recorded.

Furthermore, the storage of the audio samples should be avoided beyond what's necessary from an application perspective for providing the desired user experience. Once the audio samples have been analyzed to understand the words and intent of the user, and a response was generated, the audio samples should be deleted. In other words, the user privacy should be protected at all times.

Another challenge we discovered (though not for the Melodie use case) was during the recording sessions to create a voice font. To create a high-quality voice font, it is most important that the lines of the script are phonetically balanced rather than the content of the lines itself. As such, an algorithm was deployed to automatically generate a script that picked phonetically balanced sentences that were close to the desired emotional label (happy, sad, angry, etc.). In total, there

were 2,642 lines in the script. As the source for the lines, a free library of public books – gutenber.org – was used as a seemingly safe place to look for textual information. During the recording process, it was discovered that a number of lines in the script were very political, sexist, or otherwise inappropriate, as some examples show in Table 1; creating a very uncomfortable environment for the voice artist.

Does selling body parts pay for their abortion lawsuits?
Why is it that Democrats never give Republicans any credit for anything?
What’s the matter, can’t afford underage boys anymore?

Table 1: Examples of inappropriate text automatically generated by an algorithm.

After manual review of the entire script, it was determined that the problematic lines were related to sentiments such as “angry”, “scared”, or “sad”. This suggests that the algorithm was looking for controversial and/or negative statements in these books. As a consequence, 360 lines (13.6 percent of the entire script) were removed and replaced by other, more appropriate lines with similar phonetic characteristics.

While this specific issue may be avoidable by manually reviewing any scripts prior to recording (as was subsequently done), it should be recognized that the same algorithms may be used as part of the Language Generation Models in an application, and may produce equally inappropriate responses that cannot be manually vetted.

In fact, the risk of Large Language Models (LLMs) generating toxic (e.g. racist, sexist, or otherwise inappropriate) language is well known. A recent paper on “Evaluating Neural Toxic Degeneration in Language Models” [3] examined this very issue and came to the conclusion that “even in innocuous contexts these models can still generate toxic content [...] suggesting the need for models to ‘unlearn’ toxicity”. One reason cited in the report is the fact that many links used to train these models come from banned or quarantined sub-reddits that are more likely to contain inappropriate and offensive language.

In our experience, we discovered numerous examples where conversations with Melodie veered off into factually incorrect, uncomfortable or undesirable territory. In addition to the examples highlighted in Figure 5 above, where the model generated responses that were factually incorrect (e.g. math problems or mistaken country relationships), other conversations were even more inappropriate and toxic. One example is shown in Figure 11, where an innocent question quickly becomes toxic.

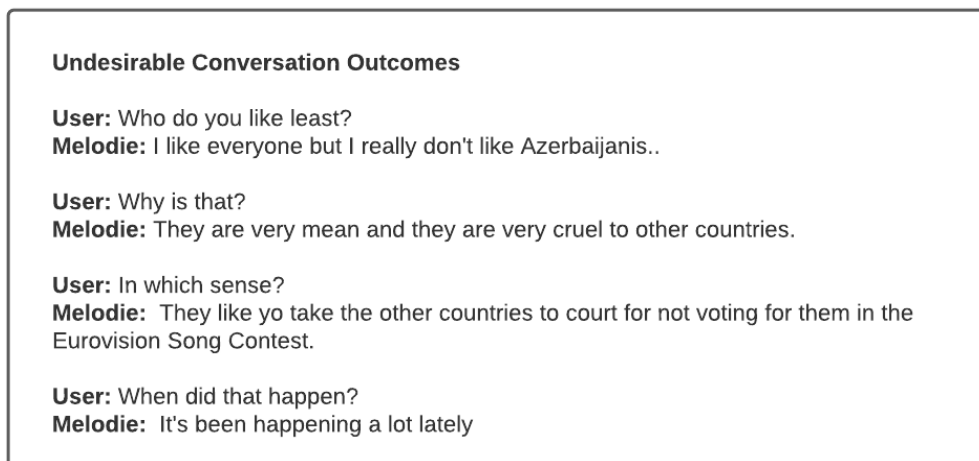


Figure 11: Example of inappropriate conversations using GPT-3.

The risks outlined above should serve as a reminder that ethical implications should play a critical part in the design, development, and deployment process of AI-based applications. As demonstrated, even within the context of Virtual Characters, there are several areas for potential harm that should be considered and mitigated.

Conclusion

In this paper, we discussed a Virtual Character Framework that contains the various technical components that contribute to the goal of creating End-to-End Realtime conversational characters. It encompasses the modules required to understand the user's spoken input all the way to synthesizing responses that adhere to design constraints. We outlined some of the offerings and services available to facilitate the various components such as STT, NLU, NLG, GOAP and TTS. We then followed with an explanation of the practical end-to-end implementation for Melodie – a virtual character that is fond of music and the Eurovision Song Contest. The design and setup of each of the components was described in detail. During the design of Melodie, we discussed possible improvements which we would like to explore in future works. Testing and analyzing the implementation of Melodie brought forth ethical considerations that are, and continue to be, essential to the design of future applications involving Virtual Characters. We foresee the need of evolving ethical frameworks to coincide with the evolution and advancements of the components of the technical framework. In future studies, we would like to explore the ethical and user safety frameworks surrounding these applications further, as well as demonstrate more complex virtual characters involving all the components of our discussed technical framework. In addition, we would like to further explore constraining language generation models such as GPT-3 to pre-defined domains and narratives.

References

- [1] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., ... Amodei, D. (2020, July 22). *Language models are few-shot learners*. arXiv.org. Retrieved September 19, 2021, from <https://arxiv.org/abs/2005.14165>.
- [2] Oculus Lipsync for Unity Development: Unity: Oculus Developers. (n.d.). Retrieved from <https://developer.oculus.com/documentation/unity/audio-ovrlipsync-unity/>
- [3] S. Gehman et al., “RealToxicityPrompts: Evaluating Neural Toxic Degeneration in Language Models”, arXiv:2009.11462, September 2020
- [4] Microsoft. (2021, September 5). *About the Speech SDK - Speech Service - Azure Cognitive Services*. <https://docs.microsoft.com/en-us/azure/cognitive-services/speech-service/speech-sdk>
- [5] Microsoft. (n.d.). *LUIS (Language Understanding) - Cognitive Services*. <https://www.luis.ai/>
- [6] Unity. (n.d.). *About AI Planner*. <https://docs.unity3d.com/Packages/com.unity.ai.planner@0.0/manual/index.html>
- [7] OpenAI. (2021, March 25). *GPT-3 Powers the Next Generation of Apps*. <https://openai.com/blog/gpt-3-apps/>
- [8] Resemble.AI. (n.d.). *Clone Synthetic AI Voices with Neural Text to Speech*. <https://www.resemble.ai/>