AI Character for Theatrical Performance

Daniel Mathew
1216989033

Ravi Bhushan
1214347783

Sharath Mohan Maidargi
1215110182

Vishal Kumar Pandey
1217402095

Arizona State University, Tempe, Arizona

December 14, 2019

Abstract

Intelligent conversational agents or chat-bots have traditionally been used for linear conversations with their users. From Slack bot to Google Assistant, Amazon Alexa, and email bots, these agents serve as an interface to interact with the software. These dialogue systems, are however mere bookkeeping and question answer systems at their core. They lack emotions and are unable to provide an engaging conversation to the user. In this paper, we propose a conversation system based on Intent Response Model. Our system can handle the non linearity of a theatrical character. We propose an AI character for Theatrical performances.

1 Introduction

Theatre is a collaborative form of performing art that uses live performers, typically actors or actresses, to present the experience of a real or imagined event before a live audience in a specific place, often a stage. An actor is a theatrical character who conveys his or her emotions through a sequence of dialogues and facial expressions. This character can interact with audience, gauge their reactions and appropriately react back. In this work, we have tried to incorporate the intricacies of a theatre actor into an AI chatbot. The conversation systems like Alexa, Google Assistant, lack the non linearity of a

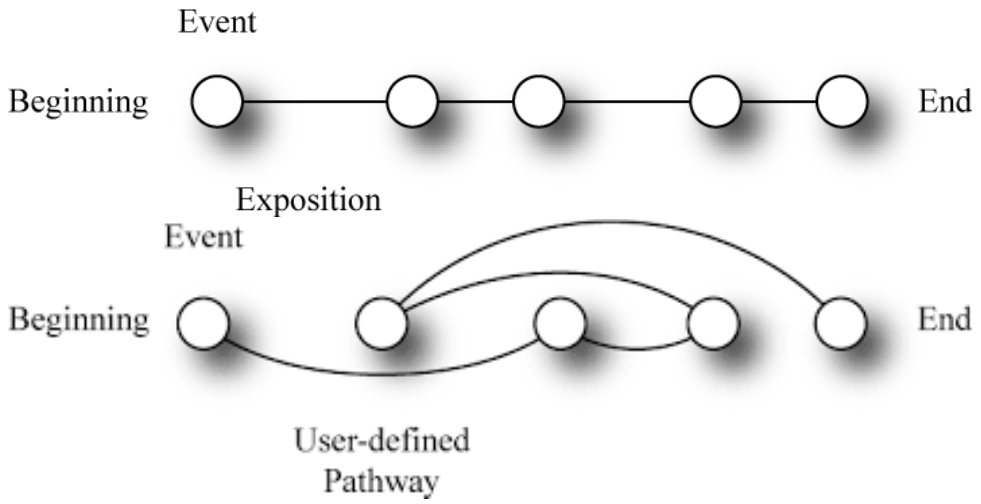

Figure 1: Linear and Non-linear Conversation

theatrical character. We propose an Intent Response architecture which will be capable of random conversations with the user. Using this architecture, one can build an AI conversation system that will have its own character and will be able to change its conversation style and content based on the emotional feedback from user. This system can also perform other non linear tasks like generating a poem and hosting a game for the audience.

Section 2 of this paper talks about previous works in this field. Section 3 will talk about the architectural overview of our proposed system. Section 4 will talk about the experimental setup. Section 5 will talk about results. Section 6 will conclude this paper and give a summary of future works.

2 Related Work

Our architecture uses Amazon Lex's[11] intent/response model to generate appropriate textual output to user's chat input. The highly restrictive environment of Amazon Lex does not support non linear conversation and cannot generate chats without the initiation of user. Our conversation system uses Universal Sentence Encoder[1] to match intents to appropriate responses. Universal Sentence Encoder uses word embedding to find similarity between sentences. We will discuss more about it in section 3.

[9] has different models for generating poetry. The main focus was sonnets, which not only have a constraint on the number of syllables per line, but must follow a specific rhyme scheme. One model was phoneme-level, which was used to constrain the structure at training time. The other, a character-level model, could produce more coherent text, but did not have any constraints on form. The text was constrained at sample time by using a discriminator that would reject text that did not conform to the desired meter.

Emotional Chatting Machine(ECM)[8] is capable of detecting textual emotions and give an emotional response in one of the emotional categories. ECM uses Bi-LSTM for Emotion classification and GRU for conversation generation.

[10] uses a standard character-level recurrent neural networks (RNN) to generate haiku. The output resembles haiku at a glance, but is not cohesive and do not follow any structure of haiku. [2] uses a rule based model for identifying the topic of conversation.[3] talks about Encoder-Decoder framework for text generation.[4] concentrates on profiling the user and generating appropriate content.

Self-Feeding Chatbot architecture [5] focuses on improving the chatbot response in real time. It uses user's input text to understand if its response was appropriate and feeds back the result to its model to generate correct output. Controllable neural text generation method [6] uses conditional training and weighted decoding, in order to control four important attributes for chitchat dialogue: repetition, specificity, response-relatedness and question-asking.

3 Architecture

The diagram below(Figure 2), gives an architectural overview of the system. Max system will control the AI character using an UDP connection. The AI character will have two modules. Chatbot module will be responsible for generating textual output. Camera Module will be responsible for profiling the users and sensing their emotions.

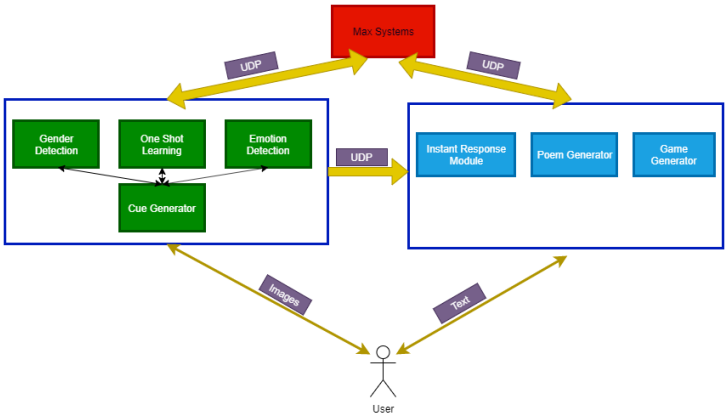

Figure 2: System Architecture

3.1 Camera Module

Camera module is responsible for profiling the user. It is a combination of different machine learning models which detect different visual characteristics of the user. Emotion Detection module detects emotions of the user. The emotions that can be detected are angry, sad, happy, neutral and surprised. Gender detection module detects the gender and age of the user. One shot learning stores user facial

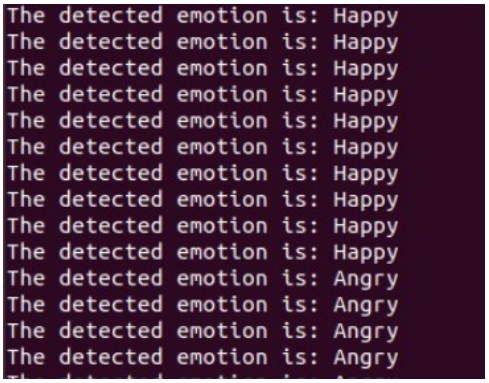

Figure 3: User Emotion Detection

information and is able to recognize the user when they return back. Cue learning scans the user for visual cues and interesting objects like flowers, hat etc. The output of Camera Module is sent to the chatbot module using an UDP connection.

3.2 Chatbot Module

Chatbot module is responsible for interacting with the user through Max system via UDP connection. It also interacts with the camera module via UDP connection. It will send and receive UDP data from the camera module. It receives the user emotional data from the camera module and uses this data to generate appropriate responses from one of its sub modules. It has the following sub modules:

3.2.1 Intent/Response Module

This module is converses with user and generates textual responses to any input text. It is a topic based chat-bot capable of generating contextual responses based on a particular topic and chat history. Data structure is used to store intents , their utterances and corresponding response/slots.

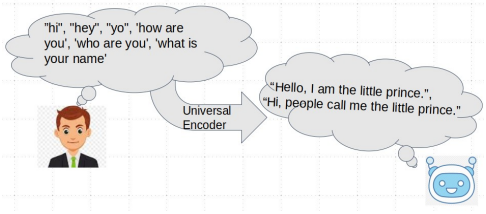

Figure 4: Intent Response Architecture

Certain intents might have actions in addition to direct response to the user interface like poem generation. To map the user input utterance of an intent to appropriate response, we use Universal Sentence Encoder algorithm [1]. This algorithm uses a special kind of word embedding to find similarity between sentences taking word order and position into account. Universal Encoder use Deep Averaging Network(DAN) which gives us a far more accurate result than AWS Lex's intent matching module.

An AI character also needs to narrate a story or a dialogue and involve the user in it. Traditional Intent Response module lack this feature of tracking the progress of a story. We modified this system by adding weights to each intent and forced the Universal Encoder to choose an intent which has higher weight than the current intent. In this way, keeping track of the progress of story will be convenient. Named Entity Recognition modules such as Spacy were used to implement slots, but did not give satisfactory results. Amazon Rekognition is being considered to identify Person, Date, Location and Time as it is far more accurate. We will take a call based on the project implementation.

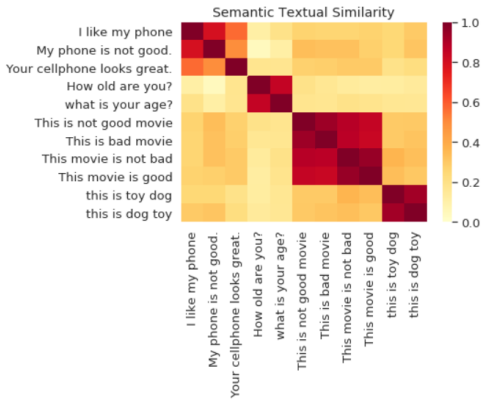

Figure 5: Universal Encoder

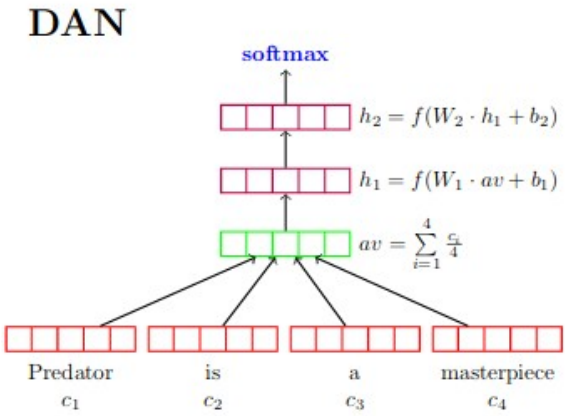

Figure 6: DAN

3.2.2 Poem Generator

This module is responsible for generating poems real time. It uses character-to-character text generation model having a LSTM network. This network was trained on Tempslibres, Sam Ballas' PoetRNN corpus and Herval Freire's Haikuzao corpus. The trained model can generate poems of different syllable lengths. It can also take in a starting word and generate appropriate poetry.

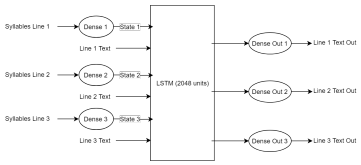

Figure 7: Poem Generator Model

Figure 8: Poem Generator Training

3.2.3 Game Generator

Game generator module will have the capabilities of engaging the audience in an interesting manner. It can ask the audience to play games based on particular set of rules. Using the camera module, it will

be able to detect if the rule has been obeyed or not. This module is still in progress.

3.3 Max System

Max system is used as an orchestrator and it is at the highest level of the architecture. It also acts as an interface between the different outside world and our system. The communication between the Max system and our system is by using the UDP (User Datagram Protocol).

3.4 Inter Module Communication

We are using UDP (User Datagram Protocol) for wireless communication between different modules. The implementation in done in python programming language. The results of the individual system is shared with the other systems using this protocol.

4 Experimental Setup

4.0.1 Theatre Setup

Figure 9 represents the theatrical setup. It consists of a series of leds whose position on stage can be controlled and hence they can create any landscape or form.

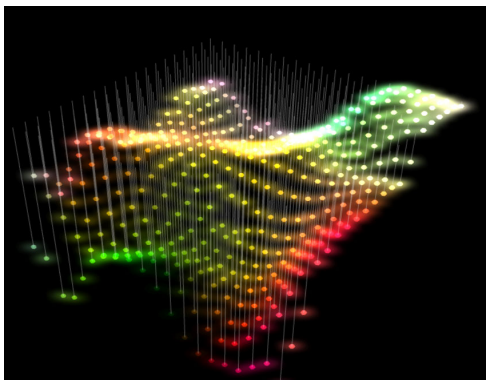

Figure 9: Theatre Model

4.0.2 Chatbot Setup

The AI chatbot interacts with the user using a mobile application. The output of the chatbot is converted to speech using the Max system and is related on to the stage in the form of sound.

5 Results

The Max system, chatbot module and camera modules are interacting with each other using UDP. The chatbot module uses Max system as an input output medium and camera system to understand the user.

5.1 Chatbot response

Figure 9 and figure 10 are sample responses of the chatbot. We gave the chatbot the character of little prince from the Children's book "The Little Prince" by Antoine de Saint-Exupéry. In the below conversation, the chatbot 'little prince' is talking about a rose flower which he tended to at his planet. As can be seen in the chat, the chatbot is able to detect the emotion of the user to be neutral. It uses the emotional state of user to generate texts that are unique to that state.

Below is a sample response of Poem Generator module. The AI character generated these poems in real time.

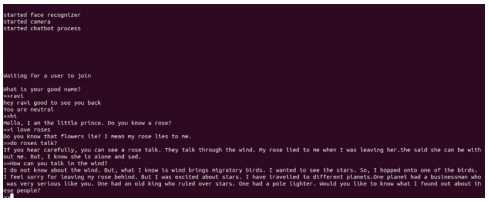

Figure 10: Intent/Response Module Output

```
sometimes i wish i
ould say i was just confused
cause the weather are


sometimes you gotta
top and that's why i forget
who i am to but
```

Figure 11: Poem Generator Module Output

The generator can also take in the length of poem to be generated and key word and output a poem in required format.

5.2 UDP Connection and User Profiling

UDP connection is used as a medium of connection between modules. Below is the response of interaction between chatbot module and camera module. The camera module waits for the user to join. Once the user joins, the module transfers the user's gender, emotional state and age to the chatbot module via UDP. The chatbot module receives this information and creates a profile of the user. It then users this user data while interacting with the user.

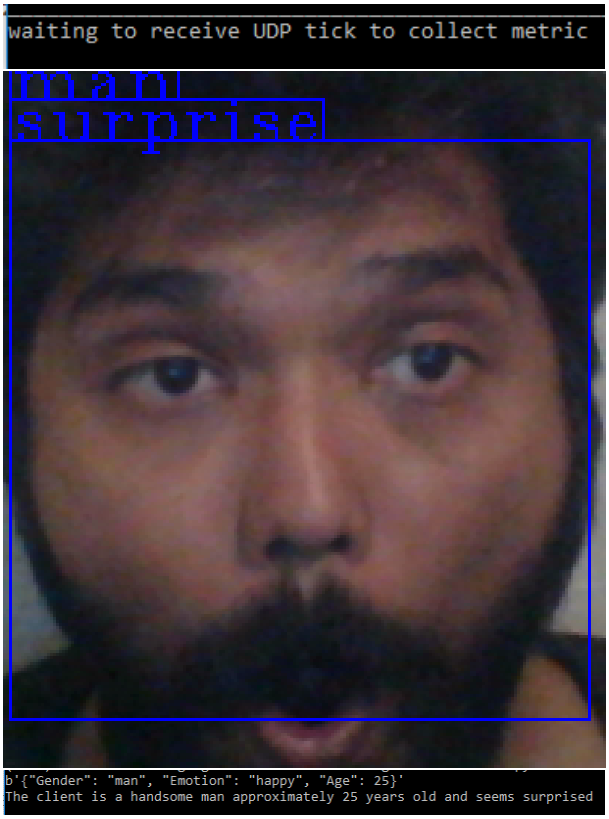

Figure 12: User Profiling via UDP

6 Conclusion

In this paper, we presented the architectural overview of a chatbot which can simulate a Theatrical character. Apart from verbally communicating with the user, the chatbot also tries to understand the non verbal user input through cameras. This makes the chatbot responses more robust and appropriate.

In verbal communication, we have introduced novel ideas of using Universal Encoder while matching intents to responses and using weighted intents to monitor story progress. Our model was better at matching intents to responses than the curent Alexa model.

As a part of future work, we propose to extent the game module. We would like to create a self learning chatbot and strengthen our emotion detection model in low lights. We also plan to detect objects like hats, flowers.

References

[1] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuanc, Chris Tar, Yun-Hsuan Sung, Brian Strope, Ray Kurzweila, "Universal Sentence Encoder", Conference on Empirical Methods in Natural Language Processing, Brussels, 2018

[2] Ben Krause, Marco Damonte, Mihai Dobre, Daniel Duma, Joachim Fainberg, Federico Fancellu, Emmanuel Kahembwe, Jianpeng Cheng, Bonnie Webber, "Edina: Building an Open Domain Socialbot with self-dialogues", Alexa Prize Proceedings, 2017

[3] Yansen Wang, Chenyi Liu1, Minlie Huang, Liqiang Nie, "Learning to Ask Questions in Open-domain Conversational Systems withTyped Decoders", Association for Computational Linguistics, New Orleans, 2018

[4] Yury Zemlyanskiy, Fei Sha, "Aiming to Know You Better Perhaps Makes Me a More Engaging Dialogue Partner", Conference on Computational Natural Language Learning, Brussels, 2018

[5] Braden Hancock, Antoine Bordes, Pierre-Emmanuel, Mazar eJason Weston, "Learning from Dialogue after Deployment:Feed Yourself, Chatbot!", Association for Computational Linguistics, Florence, Italy, 2019

[6] Abigail See, Stephen Roller, Douwe Kiela, Jason Weston, "What makes a good conversation?How controllable attributes affect human judgments", NAACL-HLT, Minneapolis, 2019

[7] N. Majumder, S. Poria, D. Hazarika, R. Mihalcea, E. Cambria, and G. Alexander, "DialogueRNN: An Attentive RNN for Emo- tion Detection in Conversations," AAAI 2019

[8] Hao Zhou, Minlie Huang, Tianyang Zhang, Xiaoyan Zhu, Bing Liu, "Emotional Chatting Machine: Emotional Conversation Generation with Internal and External Memory", AAAI Conference on Artificial Intelligence, New Orleans, 2018

[9] Jack Hopkins, Douwe Kiela, "Automatically Generating Rhythmic Verse with Neural Networks", Association for Computational Linguistics (ACL 2017)

[10] Tanel Kiis, Markus Kängsepp, "Generating Poetry using NeuralNetworks", https://neuro.cs.ut.ee/wp-content/uploads/2018/02/poetry.pdf

[11] Amazon Lex, https://aws.amazon.com/lex/