POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

An architecture to implement emphatic dialogue generation in a robot interacting with a human in improvisational settings

Laurea Magistrale in Computer Science Engineering - Ingegneria Informatica

Author: Simone Cattaneo

Advisor: Prof. Andrea Bonarini

Academic year: 2022-2023

1. Introduction

Social robots are autonomous robots able to interact both physically and verbally with a human interlocutor following social rules and conventions. The main difficulties in the development of these systems are due to the complexity of the human relationships: for this reason the research in this field has not yet reached a final and definitive result. The improvisational theatre field is the perfect environment where the social behaviors can be tested in an unconstrained setting since the stage is a space outside the rules of time and space, customizable, and in which any kind of situation can be created. The goal of this work was to implement on an existing robot, already able to interact with a human agent through its body movements, the ability to communicate verbally, having a turn-taking conversation with an interlocutor, and generating emphatic responses coherent with the context both from the conceptual and the emotional point of view.

2. Context and goals

This project can be considered a part of a broader line of research which aim is to build an autonomous robot able to interact in an improvisational setting with a human agent. In order to better understand how this work is linked to the previously developed architecture, its main features are briefly outlined. When I started to work on it, the robot, thanks to the previous contributions of colleagues [1, 4], was able to classify the human actor movement and gestures in one of a discrete set of 16 scenic actions basing on:

- the emotion, expressed by both the face and the body pose;
- the proxemic zone, which is the zone occupied by the actor with respect to the robot and determines the intimacy he or she has with it;
- the proxemic movement, determined by the direction and speed of the movement of the actor.

The robot was then able to select the proper reaction basing on a Finite State Automaton that accounted for the personality of the robot (defined in advance), the last scenic action of the human actor, and the current emotional state of the robot.

2.1. Goals

The goal this project had to achieve was to implement a module to enable the robot to sustain a conversation in an improvisational context. To

be more specific:

- the robot should be able to generate emphatic responses coherent with the emotional state of the interlocutor;
- the robot should be able to generate responses conceptually coherent with the situation in which the dialogue is grounded;
- the robot should be able to generate responses coherent with the scenic action performed by the human actor;
- the robot should be able to synchronize itself with the human interlocutor, waiting that he/she finishes to pronounce his/her sentence and/or perform his/her action before generating an answer, possibly introducing the smallest possible latency.

Because of the complexity of the task I also made two main assumptions:

- only turn-taking conversations with a single interlocutor are considered;
- the scenic actions of the actor classified by the existing architectures are always correct.

2.2. Hardware constraints

The only big constraint this project had to satisfy was to keep everything local to the robot in order to be independent from possible breaks on the internet connections that the use of online APIs may cause. This limitation mainly affects the choice of the language model to generate the responses of the robot, since many state-of-art deep learning models require a huge computational power both at training and at inference time because of their dimension. For this reason I had to select a model that fitted on the available device that was the *NVIDIA Jetson AGX Orin* module which is compact, has lots of connectors and is designed to be the largest to be integrated on robots and machines.

3. Technological solutions

In order to achieve the goals of the project, I had to deal with four different tasks that correspond to four different technological solutions:

- Text generation: this is the core of the project. The goal is to find out the most suitable model for this context and train it in order to generate proper utterances for the robot.
- emotion recognition: this module's aim is to detect the emotional state of the interlocutor from the sentence elaborated by the STT module.
- STT: the "speech-to-text" module has to convert the human speech into text, deleting, or at least limiting, the noise from the environment.
- TTS: the "text-to-speech" module has to convert a text in speech, in order to imitate the human voice and reproduce it through the speakers.

3.1. Dialogue generation

The model selected after many trials to generate the robot responses was *DialoGPT*, which is based on the transformer architecture of GPT-2 fine-tuned on a corpus of 147M conversations from Reddit [6]. The reasons of my choice were mainly three:

- it exists (exactly as GPT-2) in 4 different versions different in size and I could select the medium one with 340M parameters which was the biggest model that fitted on the available GPU;
- it is the most used model for text and dialogue generation in the literature thanks to its auto-regressive behavior;
- it is pre-trained on a dataset of dialogues and was for this reason a good starting point for my project.

3.1.1 Training phase

Since the Reddit conversations are not generally grounded in an emphatic context I decided to fine-tune the model on a proper set of data, and I chose for this purpose the *Emphatetic Dialogue Dataset* which contains around 25K open-domain one-to-one conversations each one associated to an emotional label [5]. First of all, I added to the tokenizer of the model some special tokens to separate the words belonging to the first interlocutor (which is the human actor) and the ones belonging to the second interlocutor (the robot). Then, I processed all the conversations in the training split of the dataset to bring them in a unique format: each utterance had to begin with the token identifying the speaker and had to end with the end-of-text token. I decided to consider only conversations containing exactly 4 utterances because I wanted to find a trade-off between the length of the con-

text given as input to the model and the ability (or better the non-ability) of *dialoGPT* of finding dependencies very far in the text. Moreover, the dataset contained conversations consisting of, on average, 4.36 utterances. Finally, I trained the model using *AdamW* as optimizer, learning rate $5 * 10^{-5}$, weight decay 0.05, batch size 32 and a linear learning rate scheduler with *warmup*.

3.1.2 Training results

The selected model turned out to have the best performances on several datasets with respect to other tested models different for the value of the hyper-parameters or for the optimization algorithm used. The metric used for evaluation and testing was the *perplexity*, which expresses the confidence of the model in the prediction of the text and can be defined as the exponential average negative log-likelihood of a sequence. The final model obtained the best score of 7.70 around the end of the 2nd epoch of training on the evaluation split of the same ED dataset used for the training (as can be seen in figures 1 and 2), but had good results also on other datasets (22.78 on the Daily Dialog dataset and 13.31 on the Personachat dataset).

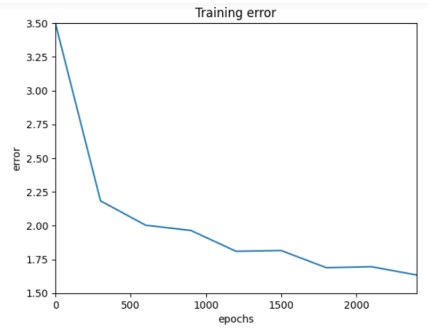

Figure 1: Training error plot.

3.2. Emotion recognition

Emotion recognition is an example of text classification with the purpose of assigning a label to a text basing on the emotion that is more or less inherently expressed by it.

3.2.1 Training phase

In order to perform this task I used *Bert* which is a bi-directional model: this feature allows it to process the whole input sequence at once and

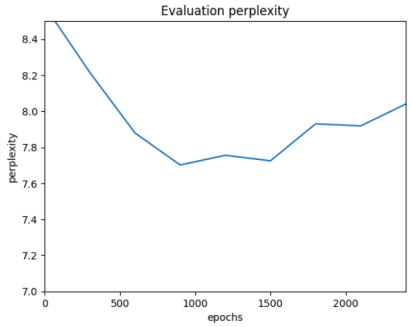

Figure 2: Evaluation perplexity plot.

allows it to better understand the context [2]. The emotion model used by the robot system is the Ekman one, which considers the existence of 6 basic emotions: anger, disgust, fear, joy sadness and surprise [3]. For this reason, I had to fine-tune the Bert model in order to classify the human interlocutor utterances in one of these 6 emotions plus the neutral one. In order to do this, I combined and processed the data from two different dataset to obtain a new dataset containing the Ekman emotion labels distributed in a proper way. Finally, I fine-tuned *Bert* using *AdamW* as optimizer, learning rate $5 * 10^{-5}$, weight decay 0.05, batch size 16, and a linear learning rate scheduler.

3.2.2 Training results

For what concerns the emotion recognition task, since it did not represent the core functionality of this project I just looked for a model with reasonably high performances: the Bert fine-tuned accuracy on the test dataset was around 85%, and it was considered enough for the purpose.

3.3. Speech To Text

Speech To Text (STT), also known as *Automatic Speech Recognition* (ASR), is the task of converting a human-like speech in textual form. It is a very important and delicate part of this project, since a bad quality transcription of the human interlocutor sentence would lead to bad response generation from the language model and to a confuse and meaningless dialogue. There were three necessary conditions to consider this task as successfully accomplished

- choose a model able to perform the STT task with high accuracy in order to pass as input to *dialoGPT* a context identical (or

at least as close as possible) to what the human interlocutor said;
- perform the task in real time: in order to introduce the smallest latency, the conversion of speech to text had to be performed live to be able to send the resulting sentence to the dialogue generation module as soon as the human ended to speak;
- remain offline: one of the constraints of the project was to be independent from any online API. For this reason, it was necessary to find a service that operates locally without using any external service.

I have exploited the services offered by *Nvidia Riva*, which is a GPU-accelerated SDK for building speech AI application allowing to easily download, customize and fine-tune state-of-art models that deal with language as well as access their functionality through APIs. For what concern my project, the main advantages were that it is optimized for embedded systems (and in particular for the Jetson GPU) and it offers the chance to easily deploy locally state-of-art models. In the end, after having evaluated the features and the performances of the main models for ASR, I decided to use the *Conformer-CTC* model through the Nvidia Riva services.

3.4. Text To Speech

Text To Speech (TTS) can be considered the opposite of STT and is the conversion of text into phonemes. To perform this task I used the combination of two tools:

- a language model locally deployed through Nvidia Riva to perform *speech synthesis*, which allows to append to the text necessary information to produce the sound, such as the spoken language, the sex of the speaker, the pronunciation...
- the python library *sounddevice* that allows to play *Numpy* arrays containing audio signals.

I didn't consider the modulation of the reproduced voice basing on the expressed emotion since it was not a core requirement of this project.

4. Software architecture

In order to implement the modules of the project I have used *ROS2*, which is an open-source middleware that can be considered a bridge between the application and the low-level details used to develop robots. Before to briefly describe the main features of each component of the architecture it is useful to have a high level view of it. Proceeding from left to right on figure 3 we can see:

- SST module to convert human speech to text;
- an interface that connects my system (that manages the verbal interaction) to the one already present (controlling the physical interaction);
- a module in charge of generating the responses of the robot in a conversational setting and of extracting the emotional state of the interlocutor exploiting a dedicated service;
- TTS module to convert the robot textual response to speech.

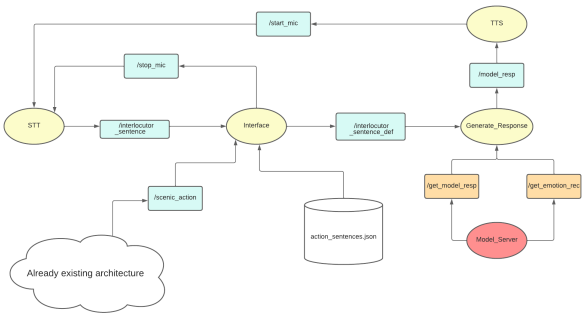

Figure 3: System architecture.

4.1. STT node

The main activity of this node is just to loop waiting for the human interlocutor starting to speak, and then to use the Riva Client library to convert the English speech into text with punctuation. In order to do that I had to face three main problems.

The first one was that the sound produced by the speakers returned in the microphone, thus producing a loop, since the robot kept answering to itself. To solve it I just implemented the possibility to start and stop the listening process of the microphone: since turn-taking conversations are only considered, after the utterance of the human actor has been detected the microphone can be deactivated and activated back only after the robot response has been reproduced.

The second problem was related to the environmental noise. The solution I adopted was just

to use a directional headset microphone and to exploit its closeness to the interlocutor mouth to set very low volume of the microphone and a low value for the minimum confidence to reject the utterance (if it is detected with a lower confidence it is considered as noise from environment and discarded).
The third problem was that every short pause of the interlocutor speech was interpreted by the Riva system as the end of its utterance and it didn't wait for the effective conclusion. In order to solve this issue, I have implemented a custom timer that is activated when the end of a sentence is detected: if in the meantime another sentence is detected the timer is reset, otherwise if nothing is detected the detected sentences are appended and the final result is published to teh system.

4.2. Interface node

The aim of this node is basically to create a bridge between the developed system and the already existing system, by exchanging information and synchronizing the verbal reactions of the robots with the body ones in order to generate verbal responses coherent both with the last sentence pronounced by the interlocutor (and generally with the context of the conversation) and with the emotions expressed through his/her gestures and expressions. Since a language model can take as input textual data I decided to map any scenic action to a string that represents it (e.g., the action sharing fear can become "I am frightened"). It also represented a challenging and interesting way to test the flexibility and the generative power of the language models. Every time a message containing the scenic action A is received, the node updates a buffer, which is a FIFO queue, of dimension N and checks if it contains at least X instances of A, where both N and X are parameters that can be configured. If the answer is *true* it is assumed that the scenic action A is effectively happening and it is not related to a mistake of the detection system, and there are two possible scenarios:

- the microphone is not active which means that the interlocutor is not speaking: the node sends a signal to stop the microphone, converts A into a string and publishes it;
- the microphone is active, which means that the human agent is speaking and performing a scenic action at the same time: the action A is saved inside the node waiting for the sentence S converted by the STT node; when it is received the action A is converted into a string, concatenated with S and published.

The mapping between scenic actions and strings is managed through a json file.

4.3. Response generation node

This node receives the final string generated by the interface node and performs two actions:

- compute the response using the fine-tuned *dialoGPT* model;
- extract the emotion from the sentence using the emotion recognition *Bert* model.

In particular, these tasks are executed by using two different services to obtain a greater scalability and modularity. The first service gets as input a string, uses the fine-tuned *Bert* model to classify it in one of the 6 Ekman emotions (plus the neutral one) and returns the result. The second service gets as input a string, create the context appending the last 3 utterances and generates the response using the dialoGPT model fine-tuned on the ED dataset.

4.4. TTS node

The aim of this node is to convert to speech the textual response generated by the *dialoGPT* model, and to reproduce it through the speakers. The textual robot response is synthesized through the Riva server services, is wrapped into a numpy array and is reproduced through the speakers using the python sounddevice library.

4.5. Scenic action generator

Since it was not possible to properly test the integration of my system with the already existing architecture because of some problems encountered during a parallel thesis project on the robot body and movement, I implemented a simple node that acts like a console to simulate the sending of scenic action messages. It just loops waiting the insertion by the user of an integer between 1 and 14: when a command is received the number is mapped to a scenic action using the information contained in a json file and a sequence of the resulting scenic action messages is published.

5. Conclusions

This thesis work is inserted in a broader research aimed at building a robot able to interact with a human actor manifesting and processing emotions both from a verbal and non-verbal point of view in an improvisational context. For what concerns my project, the goal was to implement the ability of having a conversation with a human agent generating responses that manifested empathy, coherence with the overall context and that were able to surprise and entertain the interlocutor creating a positive relationship with him.
The results that I reached are:

- the robot is able to detect when the interlocutor begins to speak and to coordinate and synchronize the activities of generating the responses and reproducing them through the speakers;
- the robot is able to dialogue with a human interlocutor in an improvisational context generating responses that show a general knowledge of the situation;
- the robot is able to manifest empathy generating responses that are adequate from an emotional point of view with the state and the mood of the interlocutor;
- the verbal dimension is synchronized and connected with the non-verbal one: the emotions are extracted from the interlocutor sentences in order to generate proper movement reactions and at the same way the scenic actions detected from the human actor movements are used by the language modules to generate adequate verbal responses.

References

[1] Claudia Chiroli. Simulation of emotional behaviour in a robot. Master's thesis, Politecnico di Milano, Milan, Italy, 2022.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[3] Paul Ekman. An argument for basic emotions. *Cognition & emotion*, 6(3-4):169–200, 1992.

[4] Lorenzo Farinelli. Design and implementation of a multi-modal framework for scenic actions classification in autonomous actor-robot theatre improvisations. Master's thesis, Politecnico di Milano, Milan, Italy, 2022.

[5] Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. Towards empathetic open-domain conversation models: A new benchmark and dataset. *arXiv preprint arXiv:1811.00207*, 2018.

[6] Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. Dialogpt: Large-scale generative pre-training for conversational response generation. *arXiv preprint arXiv:1911.00536*, 2019.