Alma Mater Studiorum · Università di Bologna

SCUOLA DI SCIENZE
Corso di Laurea in Informatica

An Evaluation Model For Speech-Driven Gesture Synthesis

Relatore:
Chiar.mo Prof.
Simone Martini

Presentata da:
Simone Faggi

Correlatore:
Prof. Stefan Kopp

Sessione III
Anno Accademico 2019-2020

A nonno Romano...

Abstract

The research and development of embodied agents with advanced relational capabilities is constantly evolving. In recent years, the development of behavioural signal generation models to be integrated in social robots and virtual characters, is moving from rule-based to data-driven approaches, requiring appropriate and reliable evaluation techniques. This work proposes a novel machine learning approach for the evaluation of speech-to-gestures models that is independent from the audio source. This approach enables the measurement of the quality of gestures produced by these models and provides a benchmark for their evaluation. Results show that the proposed approach is consistent with evaluations made through user studies and, furthermore, that its use allows for a reliable comparison of speech-to-gestures state-of-the-art models.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Non-verbal communication and, in particular, gesticulation, is a fundamental aspect of language through which to convey information in addition to what is being said. Through gestures it is possible to enhance the semantics of a pronounced term, e.g. by describing the shape or position in space of an object, but also to emphasise a word or phrase on which it is intended to give importance, e.g. by increasing the velocity of hands movements.
In person-to-person interactions, gesturing is a natural, almost instinctive, form of non-verbal communication. In recent years, efforts have focused on providing embodied agents, whether virtual or physical, with the same communication capabilities as humans. In fact, the research and development of characters with the ability to gesture while speaking (conversational gestures) is a constantly evolving area of research.
Early research in this field provided rule-based approaches capable of mapping a word or phrase to a specific gesture performed by the agent. This type of approach, although functional, has strong practical limitations mainly due to poor scalability. Thanks to the progress made in recent years in the field of artificial intelligence, many researchers are working to provide the scientific community with data-driven approaches that allow the generation of gestures in a continuous domain, overcoming the main limitation of rule-based approaches. There are many works in literature that provide machine learning models that produce gestures from speech (speech-to-gesture), each with its own characteristics. Some of those produce gestures from speech text or audio, others combining both modalities (multimodal).
As well as any artificial intelligence based model, these models are subject to two types of evaluation: objective evaluation and subjective evaluation. As

for the objective evaluation, state-of-the-art works provide, although there is no consensus, statistical measures to assess the quality of the generated gestures. These measures allow for the evaluation of produced gestures by comparing them to real (ground-truth) gestures, e.g. by verifying for correspondence in position, speed or acceleration. A major limitation of this evaluation methodology is the dependence on the audio source. Indeed, ground-truth gestures to be compared with generated ones are exclusively gestures produced by humans with a human voice. In case of integrating a speech-to-gesture model within an embodied agent that speaks with a synthetic voice, it is no longer possible to provide a measurement of the quality of the gestures produced by the model. At this point, an evaluation approach that is independent from the audio source becomes necessary. Since the objective of speech-to-gesture models, in general, is to outputs gestures that are plausible and in accordance with the inputted speech, It would therefore be interesting to directly measure this input-output correspondence, rather than comparing the output with ground-truth gestures. This kind of approach allows for an evaluation that is independently of the input audio source, overcoming the limitations of statistical analyses.

Referring to subjective evaluations, in state-of-the-art works are performed large user studies in which raters are asked to give scores to generated gestures. This type of evaluation allows researcher to get a human evaluation on their personal study, but also allows for comparison between different state-of-the-art speech-to-gestures models.

This work proposes "Evaluator", a novel data-driven approach for the evaluation of speech-to-gesture models that is based on the correlation between audio and gestures. Evaluator is a machine learning model trained in a supervised fashion that makes use of recurrent neural networks, achieving an accuracy of 91% when discriminating between "good" and "bad" audio-gestures pairs. In addition, subjective evaluations from recent state-of-the-art works are taken into account and replicated as an additional assessment of the reliability of the proposed model. Indeed, it was interesting to compare results from user studies with results from a data-driven approach, to verify whether the proposed evaluation metrics fits a human evaluation.

Chapters 2 and 3 provide the theoretical and technical background relevant to the comprehension of this work. In particular, non-verbal communica-

tion is described in Chapter 2, focusing on the importance of gesticulation in communication, and basic notions of artificial intelligence and data-driven machine learning models are provided in Chapter 3.
The most relevant and interesting works at the current state of the art are presented in Chapter 4, providing a preliminary overview of how this work differs, and what contribution it wants to provide to the scientific community. In the following chapters the requirements and specifications for the development of the Evaluator model are presented, describing the technologies used and the development environment (Chapter 5). The project design and model architecture are defined in Chapter 6, while implementation details are described in Chapter 7. The model is tested in Chapter 8 and is then used, showing application use cases, in section 8.3. Finally, conclusions are drawn in Chapter 9 and suggestions for future work are made.

Chapter 2

Non-Verbal Communication

The definition of the word "communication" given by the Oxford English Dictionary states that *"communication is the imparting or exchanging of information by speaking, writing, or using some other medium"* [4]. While speaking and writing are part of verbal communication, what the dictionary refers to "some other medium" is classifiable under non-verbal behaviours and, in particular, under non-verbal communication behaviours.

In this Chapter is investigated the role of gestures as a non-verbal form of communication, exploring how they effect everyday social interactions, teaching and healthcare.

2.1 Gestures as Non-Verbal Communication

When people communicate, they gestures. People from all known cultures and linguistic backgrounds gesture [5] and gestures is a fundamental part of languages, conveying additional information to what is being said. In particular, gestures that go along with a speech are called *co-speech* gestures or *conversational gestures* and naturally accompany all spoken language. Conversational gestures not only contribute additive information to a speech, but also have important cognitive functions for organising spoken language and facilitating problem-solving, learning, and memory [6].

Scientific research on non-verbal communication began with the 1872 publication of Charles Darwin's *The Expression of the Emotions in Man and Animals* [2]. Since then, experts have conducted abundant research regarding types, effects, and expressions of this means of communication.

Types of nonverbal communication include facial expressions, eye gaze, posture and gestures, paralinguistics (e.g. loudness, tone of voice), haptics (touch), and appearance.

2.1.1 Types of Gestures

As well as verbal communication (spoken language) is characterised by its parts (e.g., phonemes, morphemes), Mc. Neill [7] has also identified two different main types of co-speech gestures: representative and non-representative gestures. According to McNeills classification system, **representative gestures** include:

- Iconic gestures: are closely related to speech, illustrating what is being said, painting with the hands. That's why iconic gestures are also called *"illustrators"*. They depict the shape, size, action, or position of an object. The difference with other types of gestures is that illustrators are used to show physical, concrete items.

- Metaphoric gestures: give concrete form to abstract ideas. Metaphoric gestures are used to shape the idea being explained, either with specific shapes such as finger pinches and physical shaping, or more general waving of hands that symbolises the complexity of what is being explained.

- Deictic gestures: are a specific form of symbolic gestures. Deictic gestures are used to refer to the location of an object in space, that's why they are also called "pointing gestures".

Non-representative gestures, instead, refer to gestures that are used along with the speech but that are not related to any kind of semantic. This type of gestures are also called "beat gestures" and they are brief, repetitive movements that occur in rhythm with speech, serving mainly to stress or emphasise specifics words or phrases.

The main distinction between representative and non-representative gestures is that the former are linked, directly or indirectly, to the semantic meaning of the speech, while the latter serve as accompaniment to the sound or the acoustics of what is being said. This distinction also leads to another aspect to consider: while representational gestures related to semantics might be specific to a culture or language, beat gestures can be considered universal, as they play a semantic-free role and accompany the acoustics of speech.

2.1.2 Effects of Gestures

Every form of non-verbal communication plays an important role in how we relate and transfer information to others, as well as how the non-verbal behaviours are interpreted by those around us. Referring to gestures, their impact in social interactions is visible in adults but also in children behaviours. Iverson and G. Meadow [8] suggest that *"the gestures children produce when they are not yet able to speak, predict which words will enter that child's vocabulary first"*.

Teaching

During adolescence, school is where social interaction takes place most of time. Children learn from lessons held by teachers who speak to the class and usually use a blackboard where they write down some key words of what they are explaining verbally. Teachers also gesticulate as they speak, and their gestures have been found to affect children's learning. In particular, studies conducted by Valenzano et Al. [9] of classroom learning have revealed that children learn better and show better retention and transfer of new learning when their teacher gestures, while Singer [10] suggests that gestures offers learners a second message, since gestures do not always convey the same message as the speech. S. Cook [11] explains how gestures influence learning using a virtual math teacher. The author conducted experiments creating two different "virtual" teachers: they both used the same facial expression, posture and words, but one used gestures and the other didn't. Experiments show that children who learned from the gesturing teacher, learned more and more quickly. In addition, he suggests that those children generalise better their knowledge.

Healthcare

In recent years, the role of gestures in healthcare and, in particular, for cognitive communicative disorders has been explored. Cognitive-communicative disorders are deficits in cognition such as attention, memory, problem solving and information processing that also lead to communication impairments.

A recent work by S. Clough [6] investigates how the use of gestures might facilitate the uttering or understanding of communication by people with brain injury and neuro-degenerative diseases such as Aphasia, Right Hemi-

sphere Damage (RHD), Traumatic Brain Injury (TBI), and Alzheimer's Disease (AD). S. Clough, in this work, suggests that for people with RHD, who have difficulties in speaking, producing often flat or monotone speech, the use of gestures facilitates their speech utterance by lightening the cognitive workload. She also suggests that gesture plays a crucial role in promoting memory and learning. Indeed, for people affected by AD, a neuro-generative disease characterised by gradually declining abilities in learning and memory, and also in observable impairments in connected speech and language as the disease progresses, the use of gestures by those communicating with them affects memory retrieval by providing a link to experiences and knowledge, and by stimulating the processes that support the encoding, consolidation and retrieval of information by people with AD.

Chapter 3

Artificial Intelligence and Embodied Agents

The term Artificial Intelligence (AI) refers to the branch of computer science that deals with the design and development of artificial artefacts that exhibit some form of intelligence. AI, in general, focuses on building programs that try to imitate the way a living being learns new things and is a large area of research that includes multiple sub-fields (see Figure 3.1.). One of the most interesting fields is machine learning (ML), a research area that provides programs (i.e., models) with the capability to automatically gather data and learn directly from them.

Nowadays, AI models are integrated into everyday products, starting from web and mobile applications where AI is used, for instance, for personalised advertising campaigns and automated customer services, to everyday mobility, improving road safety with real-time obstacle detection systems and autonomous driving, and ending with home automation systems such as voice assistants. In a futuristic perspective, that is contemporary referring to scientific research and prototypes, AI systems also include embodied agents that through the use of AI display human behaviours, act like humans, but also interact with them and within the environment.

In section 3.1 a brief theoretical and technical background for ML is given. Then, in section 3.2, embodied agents are presented, focusing on the importance of the design for these systems and how it is carefully studied to improve their interaction with humans.

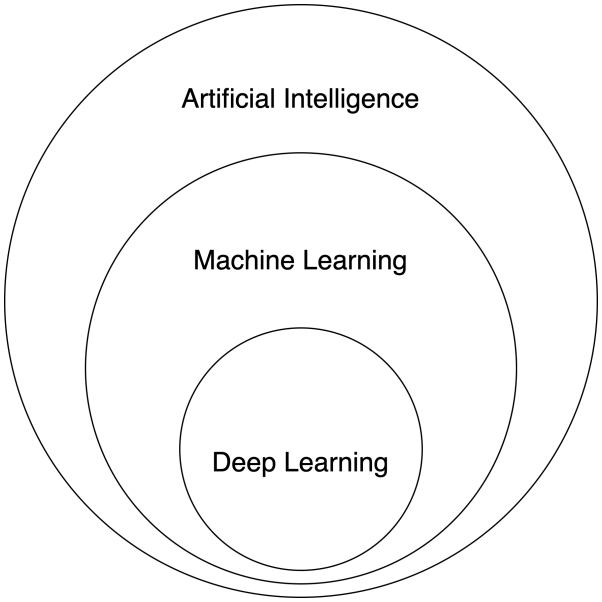

Figure 3.1: Artificial intelligence and its sub-fields.

3.1 Machine Learning

Machine learning refers to any technique that focuses on teaching the machine to learn statistical parameters from a large amount of training data. One particular type of machine learning is artificial neural networks (ANN), which learn a network of nonlinear transformations that can approximate very complicated functions of wide arrays of input variables.

From the technical perspective, an ANN can be seen as a complex, black-box, function that learn to transforms input into a meaningful output. When the learning is based on example input-output pairs, it is called supervised learning (ML Supervised). In this case, ML depends on the *supervisor* (i.e., the programmer) who chooses the database of examples (dataset). The training dataset is a collection of n examples. Each example is described with a vector x_i of j features and a label y_i indicating the class it belongs. In supervised ML, there are two main different models:

- **Discriminative models** give in output the probability that the input data belongs to a specific class. The output produced by discriminative models is in a discrete space, discriminating between different kinds of data instances. A common use cases, for example, is object recognition. Given an image, the models state whether in that image there is the

object or not. In this case, there are two classes, and the model is called binary classification model.

- **Generative models** instead, produce values in a continuous space and are used to generate new data instances.

Formally, given a set of data instances X and a set of labels Y:

- Discriminative models capture the conditional probability p(Y | X).
- Generative models capture the joint probability p(X, Y), or just p(X) if there are no labels (unsupervised learning).

The simplest structure of an ANN consists of an input and an output layer with a layer in the middle (hidden layer) in which each artificial neuron is connected to the others in the successive layers (see Figure 3.2). This simple type of ANN is called feed-forward neural network.
Starting from this simple architecture, researcher designed a huge amount of different architectures that can be used in specific situation and for specific use cases. More complex and widespread ANN architectures are:

- Convolutional Neural Network (CNN): provides a scalable approach to image classification and object recognition tasks, leveraging principles from linear algebra, specifically matrix multiplication, to identify patterns within an image.
- Generative Adversarial Network (GAN): is a generative architecture that makes use of two model. A Generator to generate new plausible examples from the problem domain, and a Discriminator, that is used to classify examples as real (from the domain) or fake (generated).
- Recurrent Neural Network (RNN): which is a commonly used architecture for problems that require learning not only from the current state, but also from past events. Two main fields of use are natural language processing (NLP) and speech recognition.

Dealing with sequences of gestures, the position of the hands at a given instant is influenced, among others, by the position they had at the previous instants. Thus, when having to deal with such data, it is common to find in literature the use of a RNN architecture.
In the next section (Section 3.1.1) characteristics and potentialities of RNNs are discussed in detail.

3.1.1 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) differ from feed-forward networks, in which artificial neurones can be only connected to next layers (unidirectional), in that the connections between layers can be bidirectional, allowing each neuron of the network to be connected not only to the next layer but also to the previous one. From a technical perspective, this distinctive property of RNNs introduces the concept of **network memory**. Thanks to the bidirectional connection, the output of a neuron can influence itself, in a subsequent time step, or it can influence neurones of the previous chain that in turn will affect the behaviour of the neuron on which the loop is closed. Figure 3.2 shows a visual representation for a RNN and a comparison with feed-forward neural networks.

The bidirectional architecture allows to dealing with short-term dependencies. For example, predicting the final word in the phrase *"The colour of the sky is ..."*, RNNs do not need to remember what was said before this, or what was its meaning, all they need to know is that in most cases the sky is blue, having to remember, in this case, 5 previous words (short-term memory). However, simple RNNs fail to understand the context behind an input (long-term memory). Something that was said long before, cannot be recalled when making predictions in the present. To address this, there are more complex architectures of RNNs which have been proposed. Among the best known are those based on Long Short Term Memory (LSTM) and Gated Recurrent Units (GRU).

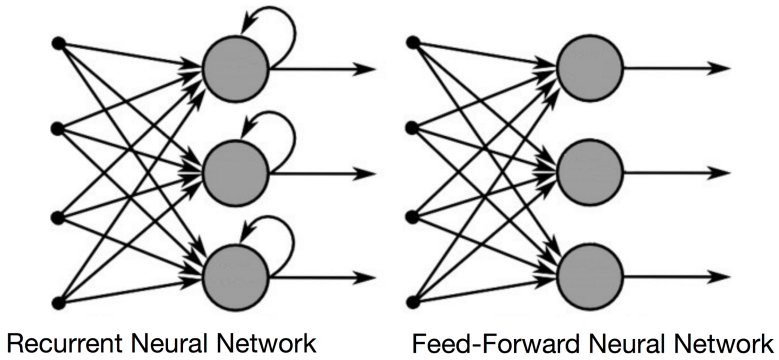

Figure 3.2: RNN vs. FFNN. Image by [1]

Long Short Term Memory

Long short-term memory (LSTM), presented by S. Hochreiter and J. Schmidhuber in 1997 [12], is a technique that has contributed significantly to improving the development of artificial intelligence.
The LSTM method solves the long-term memory problem by using three gate types for an LSTM cell for better recall: An Input Gate, a Forget Gate and an Output Gate. In this way, LSTM, unlike conventional RNN, enables a kind of memory of previous experiences: a short-term memory that lasts for a long time.

Gated Recurrent Unit

Gated recurrent unit (GRU) is a gating mechanism in RNNs published by Kyunghyun Cho et al. in 2014 [13]. GRU is similar to LSTM with forgetting gates, but has fewer parameters than LSTM, resulting in a faster trainable model. Moreover, the performance of GRU in certain tasks of polyphonic music modelling and speech signal modelling has been shown to be similar to that of LSTM; GRU has been shown to perform even better on certain smaller data sets.

3.2 Embodied Conversational Agents

Embodied Agent (EA) is a term that refers to either a physical or a computed-generated virtual character that exhibit human-like appearance (i.e., humanoid), and also displays human-like behaviour while interacting with people and its environment. In particular, when referring to embodied agents that interact with humans by speaking, they are called Embodied Conversational Agents (ECA). In the last decade, the design and development of ECAs has become increasingly popular, trying to make them conveying more and more human-like behaviours.
M. Thiebaux in 2008 presented *SmartBody* [14], a framework based on keyframe interpolation, motion capture and procedural animation for real-time animation of virtual ECAs. A more recent work by H. Tanaka [15] proposed a multi-modal framework to improve the empathic capabilities of ECAs, allowing for a socio-emotional behaviour and smoother interactions.

In general, thanks to the ability of ECAs to look human both aesthetically and behaviourally, their use can range from companion agents, to agents

who make services available by relating "humanly" to human customers, greatly enhancing their interaction capabilities. It is this latter feature that distinguishes ECAs from other interactive softwares.

3.2.1 Human-Robot Interaction

A central point in developing objects and programs that human use to interact with, is to design them in order to provide usable products, i.e. easy to learn, effective to use, and enjoyable to experience with.
There are many objects with which people interact every day. From electronic ones like smartphones, coffee machines and remote control, to mechanicals, like a door handle, steering wheel and screwdriver. For each of them, the design has been studied taking into account the purpose for which they are built and trying to put the end-user in the best use conditions. Referring to these objects, their design is examined in depth to satisfy every characteristic and to increase ergonomics, i.e., to make the objects as comfortable and efficient as possible for the consumer. This is also true for software, for which the study of user interface and user experience (UI/UX) is increasingly becoming popular in recent years, trying to provide consumers with applications, websites and, generally, programs that are easy to use and intuitive to interact with.
In the last decades, research is also being carried out into how robotics[1] systems should be designed and built to interact with humans.
Human Robot Interaction (HRI) [16], is the field of study dedicated to understanding, designing, and evaluating robotic systems for use by or with humans. As early as in 1970, Masahiro Mori defined the Uncanny Valley theory [17]. It describes the effects of the appearance and the movement of the robot on the HRI. According to Masahiro Mori, there is a threshold where robots resemble both humans and robots and it's easy to get confused in categorizing them. Instead, if the appearance of the robot exceeds that threshold, the quality of the interaction improves considerably. About HRI, W. Chung [18] suggests that, since robots are becoming more and more prominent in our society, the need for these systems to adapt to humans becomes more and more important; while B. Baumgaertner [19] states that: *"As humans anthropomorphize robots, an empathetically interacting robot is*

[1] In the following, reference will be made to robots in particular, but the same concepts can be extended to any other embodied agent designed to interact with humans.

expected to increase the level of acceptance of social robots". In addition, a work by Bailenson et Al. [20] suggests that, when creating AI humanoid systems designed to interact with humans, it is important to generate naturalistic looking gestures that are meaningful with the speech.
In general, it seems that humans prefer human-like robots rather than "robot-like" robots not only in terms of physical characteristics but also in terms of their behavioural actions. Thus, the development of an ECAs with the ability to convey information through speech with the addition of the use of co-speech gestures, greatly improves the HRI.

Chapter 4

State Of The Art

During the last few decades, research in the filed of multimodal behaviour generation and, in particular, of gesture generation, has intensified considerably. Initially, most of the methods were rule-based, but recent state-of-the-art methods are data-driven, and researcher from all over the world released a variety of so-called speech-to-gesture (S2G) models for the generation of hands gesticulation. S2G models are machine learning (ML) generative models that produce a sequence of gestures in accordance with the inputted speech, whether audio or text. Such models are developed with the aim of generating human-like gesticulations based on the semantics and/or acoustics of speech.

In this chapter most recent and relevant state-of-the-art research on S2G models are presented, discussing measures with which these models are currently evaluated, and then introducing what's the original contribution of this work.

4.1 Speech-to-Gestures Models

Early models of data-driven gesture generation are mostly characterised by the use of audio as a speech representation. Such models capture the acoustics of speech, which facilitates the production of co-speech gestures that are in accordance with the speech acoustics. Other models, on the other hand, use text as a representation of speech. The use of text facilitate the learning of gestures related to the semantics of the speech and, thus, the production of iconic, metaphoric and deictic gestures. Although text allow to retrieve important information from textual representation, they may lack

in reflecting the natural and strong link between acoustics (e.g. intonation, intensity) and gestures [21]. Most recent works, to address the limitations of both methods, propose multimodal gesture generation models, which make use of both audio and text as a representation of speech.

Text-driven Gesture Generation

Yoon et Al. [22] developed a co-speech gesture generation model from a text representation of the speech. Their model learned to produce iconic, deictic and beat gestures from TED Talks [23] speeches, demonstrating their results with the use of a social robot. C. Ishi [24] generated hand gestures from text creating word concepts using WordNet [25] and gestures classes (e.g. iconic, beat) through a clustering analysis, and then mapping the speech text to the corresponding hand gestures.

Audio-driven Gesture Generation

Most prior work on data-driven gesture generation has used the audio-signal as the only speech-input modality in the model. Hasegawa et Al. [26] proposed a gesture generation model based on a bi-directional LSTM Network. They make the use of a RNN architecture to learn *"speech-gesture relationships with both backward and forward consistencies over a long period of time"*. In 2019, after proving the importance of representations for speech-driven gesture generation [27], T. Kucherenko [28] presented a novel framework for automatic gesture generation from row audio which makes use of representation learning through autoencoders. That same year, Ginosar et al. [29] developed a speaker-specific gesture generation model which makes use of convolutional neural network to generate 2D poses from spectrogram audio features.

Multimodal Gesture Generation

Referring to multimodal gesture generation, there are few relevant recent works. The one by C. Chiu [30] presented a deep learning approach for the prediction of 12 co-speech gestures classes. In 2020, T. Kucherenko [31] developed *"Gesticulator"* a multimodal S2G model that takes both audio and text representation of the speech as input. In distinction to the work of C. Chiu, this latter work aims to produce arbitrary gestures as a sequence of 3D poses instead of a discrete gestures class.

4.2 Evaluation of Speech-to-Gesture Models

At the current state-of-the-art, S2G models are evaluated conducting both an objective and a subjective evaluation study. As for the objective evaluation, gestures produced by the model are compared with ground truth (GT) gestures using statistic measures. In order to compute these statistics, gestures must be in 3D joints positions representation. In general, there is no consensus in which statistic to use. However, commonly used statistic measures in above cited works are:

- Average Position Error (APE): is the average difference over all frame for all joints positions between GT and predicted gestures.
- Average Velocity (AV): is the averaged velocity over all frames for all joints.
- Average Acceleration (AA): is the averaged acceleration over all frames for all joints.
- Average Jerk (AJ): is the averaged jerk over all frames for all joints.
- Histogram of Moving Distance (HMD): shows the velocity/acceleration distribution of gesture motion.

In general, these statistics measures provide a baseline to evaluate whether predicted gestures have a similar statistic distribution of GT gestures.

Subjective evaluations for S2G models, instead, are user studies in which humans raters evaluate predicted gestures in a visual representation (i.e. video). Usually, human raters are asked to answer different questions, each reflecting different aspects of gestures. For example, they might be asked to give a score on the human-likeness, on the semantic coherence with the speech (e.g. when a character says "high", a "hand-raising" gesture is expected), on the utility of gestures and also on the synchrony between character's voice and hands movements.

4.2.1 GENEA2020: An Evaluation Benchmark For S2G Models

T. Kucherenko, a PhD student from KTH university in Stockholm, is one of the most involved researcher in the field of generating non-verbal behaviour

for embodied conversational agents. As he developed several gesture generation models, many other researcher are constantly working to produce more and more innovative ones. However, each research group works individually, on its own datasets and using different visualisations tools and evaluation methodologies. To address this, T. Kucherenko recently organised the GENEA2020 workshop [3], a challenge on the generation and evaluation of non-verbal behaviour for ECA. The challenge requires participants to produce models using a common dataset, which are then evaluated by conducting a large user study. This allows to compare recent approaches with each other and to investigate the state-of-the-art in the field of multimodal behaviour generation.

4.3 Original Contribution

As described above, the field of generating non-verbal behaviour for embodied conversational agents has recently been very active and constantly evolving. S2G models are designed and developed to be integrated in embodied conversational agents in order to make them performing natural co-speech gestures according with the uttered speech. Such integration, will allows to improve ECAs interaction capabilities and their acceptance by humans.
It might be reasonable to think that embodied conversational agents may use synthetic voice, for example a robotics voice, while speaking. In this case, when evaluating predicted gestures, it will be not possible to use statistic measures (presented in Section 4.2) that compare GT gestures and predicted ones. This is because dataset used for training are provided only with gestures performed by humans using human voice and not with gestures produced by a robotic voice. To address this, the need of a GT gestures independent measure becomes necessary. In particular, it will be useful to have a measure that do not make use of GT human gestures but that is based on the correlation between inputted speech and predicted gestures, describing how good this correlation is. Such a measure allows to evaluate S2G models whatever is the audio-source and also give a direct measure of what the model wants to learn, which is the appropriateness of gestures in relation to the speech.

This work aims to provide an easy and ready-to-use solution for the presented problem by proposing a novel approach for the evaluation of S2G

models that is based on the correlation between speech and gestures. In particular, provides a ML model that takes an audio and a sequence of gestures as input and gives in output a score based on their correlation.
Such a measure also allows to have a common evaluation metrics in order to compare state-of-the-art speech-to-gesture model. An example of this use case is shown in Section 8.3.

Chapter 5

A Novel Approach For The Evaluation Of S2G Models

5.1 Overview

Research studies are moving forward to produce more and more reliable ECAs-Human interactive systems in recent year. In Chapter 2, it was argued that one of the main characteristics that allows for good interaction is the production of non-verbal communication by ECAs and, in particular, of co-speech gestures. Thus, a multitude of S2G models were developed and released from and for the scientific community in order to make progress in this research field.
However, each research team design and develop their S2G model in their own environment, using own dataset, own visualisation tools and own evaluation metrics. Referring to evaluation metrics, currently used evaluation metrics do not allow for a direct evaluation on the correspondence between predicted gestures and uttered speech.

This work presents a novel approach for the evaluation of speech-to-gesture models, proposing a binary RNN-based classifier model (Evaluator) trained in a supervised fashion that associates to an *<audio, gestures>* pair a score between 0 and 1 indicating the quality of their correlation. The higher the score, the more consistent the generation of gestures from a S2G model is with the inputted audio. By obtaining scores for a relevant amount of *<audio, gestures>* pairs produced by a S2G model, it is possible to draw an overall assessment on the quality of gestures produced by that model.

Audio	Gestures	Label
Audio_1	Gestures_1	0/1
Audio_2	Gestures_2	0/1
...	...	0/1
Audio_n	Gestures_n	0/1

Table 5.1: Demonstration of the new labelled dataset.

5.2 Requirements and Specifications

The Evaluator model is a model trained in a supervised fashion, setting in which the dataset used must contains labels to be used as the optimal target for the output. It is therefore needed a dataset containing three values for each entry:

- Audio: audio file for the n^{th} speech.
- Gestures: motion file for the n^{th} speech.
- Label: a label that states whether the pair $<audio^{n^{th}}, gestures^{n^{th}}>$ is correlated or not.

A demonstration of the required dataset is shown in Table 5.1.

Having a dataset with these specifications, it is then possible to select audio and gestures features that better represent the correspondence between audio and gestures. In order to do so, a study on audio-gestures peaks correspondence is presented in Chapter 6.
The technical implementation for building the dataset and for the feature extraction phase should be scalable in order to allow future works to simply select different features and to train the same model architecture with different objectives.
The final model architecture was built starting from current state-of-the-art S2G models. Since these models try to learn the audio-gestures correlation predicting gestures in accordance with the audio, it is worthy to start from these architectures and then work on them in order to find the optimal one.

5.3 Technologies and Environment

5.3.1 File Formats

BVH - BioVision Hierarchical data

BVH is a common used file format for motion data. It is structured in two parts:

- Header: describes the skeleton, its hierarchy (e.g., left hand is the "child" of left elbow) and its initial pose.
- Data: contains the actual motion data, i.e., the rotation value of each joint for each frame.

Listing 5.1: BVH file example.

```
HIERARCHY
ROOT Hips
{
   OFFSET  0.00 0.00 0.00
   CHANNELS 6 Xposition Yposition Zposition Zrotation Xrotation
      Yrotation
   JOINT Chest
   {
      OFFSET   0.00 5.21 0.00
      CHANNELS 3 Zrotation Xrotation Yrotation
      JOINT Neck
      {
         OFFSET   0.00 18.65   0.00
         CHANNELS 3 Zrotation Xrotation Yrotation
         JOINT Head
         {
            OFFSET   0.00 5.45 0.00
            CHANNELS 3 Zrotation Xrotation Yrotation
            End Site
            {
               OFFSET   0.00 3.87 0.00
            }
         }
      }
    }
}
MOTION
```

```
Frames:   2
Frame Time: 0.033333
 8.03  35.01   88.36  -3.41
 7.81  35.10   86.47  -3.78
```

WAV

WAV files contain digital audio data. These files have a huge size due to its uncompressed digital audio content, but contain high quality audio. The audio data contained in WAV files are also called waveforms, and these waveforms can be implemented with various bit rates and sampling rates.

CSV

Comma-Separated Values (CSV) is a text-based file format used for importing and exporting (e.g. from spreadsheets or databases) a table of data. In this format, each row of the table (or database record) is normally represented by a line of text, which in turn is divided into fields (the individual columns) separated by a separator character, each of which represents a value.

5.3.2 Development Technologies

Python

Python is a high-level programming language, first publicly released in 1991 by its creator Guido van Rossum. It is a practical, easy to use and portable language, and has an extremely rich built-in and third-part library. This latter characteristics makes Python a multi-purpose programming language to be used, among others, for web development, desktop applications, game and 3D graphics, scientific and numerical computing, data management.
In recent years, the Python programming language has seen increasing use in the development of projects related to artificial intelligence and machine learning.

Anaconda Package Manager

Python libraries and their packages give programmers the ability to reuse and, if needed, extend the work done by others. Anaconda [32] is a Python

distribution with the objective of simplify the management of python packages. It is equipped with two package managers: *pip* and *conda*. They allows, with very simple code, to create working environments (e.g., deployment and release), install third-part libraries in it and import any function from those. Demonstration code is shown in Listing 5.1.

Listing 5.2: Python example

```
# Install numpy package
conda install numpy OR
pip install numpy

# Uninstall package
conda uninstall numpy OR
pip uninstall numpy

# import numpy package
import numpy as np

# use numpy to create an array
new_array = np.array([0, 1, 2])
```

5.3.3 Python Packages

Here are briefly introduced most relevant python libraries used in this work, grouped by purpose of use.

Data Management

- **NumPy** is a Python package for scientific computing. It is a useful package when having to handle arrays and having to apply fast and complex mathematics operations on the data they contains.

- **Pandas** is a library that provides high-performance data analysis tools for Python. It allows to explore, clean and process tabular data like tables and, then, dataset.

Feature Extraction

- **bvhtoolbox** package provides functions for manipulating and converting BVH motion capture files. In this work it is used to deal with

gestures files converting BVH file into CSV table, and transforming joints' rotation angles into joints' 3D position coordinates.

- **librosa** is a Python package for music and audio analysis. In particular, it is useful for dealing with audios in wav format and allows to extract features such as spectrogram and pitch from them.

- **PyReaper** is a python wrapper for REAPER (Robust Epoch And Pitch EstimatoR). REAPER is a speech processing system that allows to estimate voicing state (voiced or unvoiced) and fundamental frequency (F0).

Training

- **Keras** is a machine learning tool for Python that runs on TensorFlow [33]. It enables for fast machine learning models training and development by providing a multitude of ready-to-use tools.

5.3.4 UniBielefeld TechFak Cluster

The development of machine learning models requires great computational power, both in data pre-processing and data processing phases as well as in the training of the model. For long-running, computationally intensive processes, the Technical Faculty (TechFak) of the Bielefeld University has a cluster of machines with significantly more computing power than normal workstation computers.
Thanks to the access permission given by the Bielefeld University, it was possible to develop the project within a reasonable timeframe and without the need to use paid cloud services.

Chapter 6

Design and Architecture

In this Chapter is described in detail the design of the Evaluator model. In Section 6.1 a new dataset is built in order to fits the data structure described in Chapter 5. Then, in section 6.2 a study on audio and gestures peak correspondence will leads to the choice of best features to be selected and extracted. The research for the binary classification model architecture is described in Section 6.3. Finally, in Section 6.4, is presented the overall pipeline of the project.

6.1 Dataset

As for requirements described in Chapter 5 the ideal dataset contains audio and gestures for a relevant number of speeches and also contains labels that state whether each pair has a good correlation or not. In order to build such a dataset, it is advantageous to start from an available speech-gestures dataset, in which each audio has its related sequence of gestures. All the entries of this dataset will be labelled as "Correlated" (1). Then, data are shuffled in order to create wrongly associated audio-gestures pairs. These new pairs will be labelled as "Not Correlated" (0).

The dataset from which the new labelled dataset is built, is the one provided by GENEA2020 [3] (see section 4.2.1). It contains:

- 30 correctly associated pairs <audio, gestures> .
- About 10 minutes each.
- Resulting in about 5 hours recording.

Audio	Gestures
Recording_1.wav	Recording_1.bvh
Recording_2.wav	Recording_2.bvh

Table 6.1: Demonstration of the original dataset by GENEA2020 [3].

Audio	Gestures	**Label**
Recording_1.wav	Recording_1.bvh	**1**
Recording_2.wav	Recording_2.bvh	**1**
Recording_1.wav	Recording_2.bvh	**0**
Recording_2.wav	Recording_1.bvh	**0**

Table 6.2: Demonstration of the new labelled dataset.

As for data formats, audios for each speech utterance are in *WAV* format, while gestures files are in *BVH* format.

Starting from the GENEA2020 dataset the new dataset is built as follows:

- Each of the 30 entries in GENEA2020 dataset is labelled as **1**, i.e., as correctly associated, or **"Correlated"**.
- Then, all the entries are shuffled in order to create the same amount (30) of wrongly associated <*audio, gestures*> pairs.
- Label them as **0**, i.e., as **"Not Correlated"**.

Therefore, the new labelled dataset contains 30 correctly associated pairs labelled as "1" and 30 wrongly associated pairs labelled as "0", resulting in a balanced dataset containing about 10 hours of speech-gestures pairs.

Table 6.1 is a demonstration of the original GENEA2020 dataset, and Table 6.2 is a demonstration of the new built labelled dataset.

6.2 Feature Selection and Extraction

The feature selection and extraction step is a crucial phase that allows for dimensionality reduction, that is discard non-relevant features, select the ones that contains most relevant information (feature selection) and, finally, manipulate selected features in order to extract those information.
Thanks to the large amount of state-of-the-art work on the use of audio features to produce gestures, it was possible to make a pre-selection of features

and, therefore, to conduct this study by considering the most commonly used features in these works for audio and gestures.

Audio Features

- Pitch: is the fundamental frequency of a musical note or sound that is perceived, and is one of the main characteristics of a sound. Pitch is the feature that makes it possible to distinguish whether a sound is high or low and depends on the frequency of the sound wave that generated it.

- Fundamental frequency (F0): refers to the *"approximate frequency of the (quasi-)periodic structure of voiced speech signals"* [34]. The oscillation originates from the vocal folds, which oscillate in the airflow when appropriately tensed.

- Mel Frequency Cepstral Coefficient (MFCC): is a coefficient representing the short-term power spectrum for speech representation based on human audio perception. MFCC features are a widely used feature in automatic speech or speaker recognition.

Gestures Features

- Velocity: represents the time rate and direction of an object's movement. Note that velocity differs from speed, since velocity is a vector while speed is a scalar value representing the time rate at which an object is moving along a path.

- Acceleration: is a vector quantity that represents the variation of velocity in the unit of time. In differential terms, it is equal to the derivative with respect to time of the velocity vector.

Considering this subset of features, it was performed a manual study on the correspondence between audio-gestures features with the objective to find out which of them better represent the audio-gesture correlation. Then, it was conducted a study on the correspondence between peaks in audio frequency and hands movements. Indeed, when people gesticulate, it is fair to assume that there is a relationship between an audio frequency peak in the stress or emphasis of a word or phrase and a peak in the hands movements,

such as velocity and acceleration. This assumption will be proved later.

First, a cross-correlation study showing the relationship between features over time gives an overview of which audio and gestures features are most related to each other. The correlation between features is represented by the heatmap in figure 6.1. It shows that the best-related features are "F0" for audio and "Velocity" for gestures. However, it also shows that there is not a good relationship between those features over time or, better, that there is not a super synchrony in values oscillations.

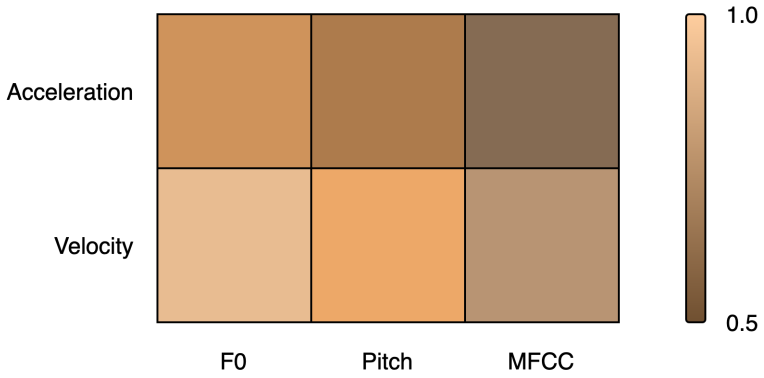

Figure 6.1: Heatmap representing features cross-correlation. 0 means bad correlation; 1 means high correlation.

This may be due to the fact that, when a speaker stress a word or phrase, there is a time delay between the time of the increase or decrease of the audio frequency and the reflection of that audio oscillation on hands velocity.

To prove the assumptions made, it is needed to check whether the correlation between audio frequency F0 and hands velocity exists, and, to check if their relationship is influenced by a certain time delay.

Start from a visual representation of the audio and gestures features over time can be useful to check whether there is a certain pattern in audio frequency and hands velocity values variation. In figure 6.2 are shown audio and gesture features of a 20 seconds speech:

- Audio: represented in green, is the F0 frequency for each frame.
- Gesture (velocity): represented in yellow, is the mean between left and right hands velocity for each frame.

- Gesture (acceleration): represented in red, is the mean between left and right hands acceleration for each frame.
- Gesture over MFCC: represent the mean hands velocity (yellow) overlaid on the audio MFCC.
- Note that, for each feature, it is also represented with a cross ("x") when a peak occurs.

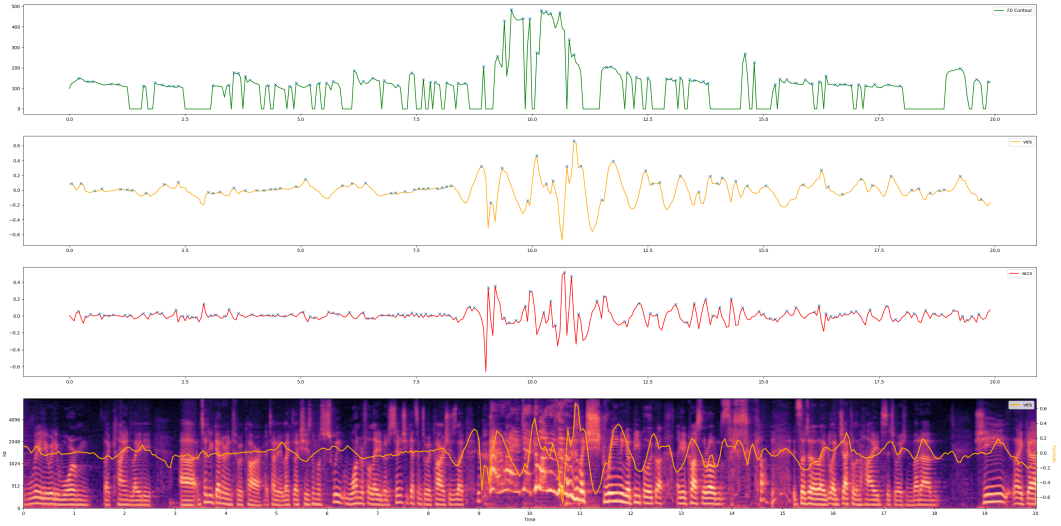

Figure 6.2: Demonstration of the study on peaks correspondence.

Looking at the plots in figure 6.2, there seems to be a pattern on how the fluctuations in audio frequency and hands velocity vary, although the results from the cross-correlation study (heatmap) were not satisfactory in this sense. Before compute the peaks delay, i.e., the time delay between a peak that occur in the audio frequency and a peak that occur in hands velocity, a formal definition of what is a "peak" should be given.

Definition 6.2.1. Peaks are indexes in which there is a (positive) variation of the function (audio/gesture feature over time) under consideration.
In particular, a peak or local maximum is defined as any sample whose two direct neighbours have a smaller amplitude. The minimal horizontal distance in samples between neighbouring peaks is 200ms.

The definition 6.2.1 is the one used in this study. However, it is possible to make deeper studies on this definition, e.g. by tuning the neighbouring parameter and/or adding other parameters like:

- Threshold: required threshold of peaks. The vertical distance to its neighbouring samples.
- Min/Max height: required height of peaks. For example, considering only peaks over 0,5 cm/s for the hand speed.

To prove the assumption that peak-to-peak correspondence may be subject to a delay and to compute that delay, a peaks delay study was performed considering:

- 5 correctly associated audio-gestures pairs.
- 5 wrongly associated audio-gestures pairs.

What is expected, is that the delay has a low value for correctly associated pairs and an high value for wrongly associated pairs.

In Figure 6.3 and Figure 6.4 is shown a demonstration of the result for one correctly associated pair and one wrongly associated pair respectively. Results show that the relationship between audio and gestures peaks has a mean delay about 0.5 seconds for correct associations and about 4 seconds for incorrect ones.
In conclusion, this analysis shows that the **F0** frequency and the mean **velocity** of the hands have a good time synchrony, even if it is subjected to small delays. Therefore, these features are extracted in order to be used as input for the evaluator model.
Another important aspect to be considered when extracting features, is that each gesture (at each frame) is influenced by the *context*. In other words, it not only depends from the audio frequency, but also from the characteristic of previous and following gestures. In the next paragraph is explained in detail the use of context, while further details on the feature extraction implementation are given in Chapter 7.

The Use of Context

A single gesture, at instant t, is influenced by the gesture immediately preceding $(t-1)$ and influences the gesture immediately following $(t+1)$.

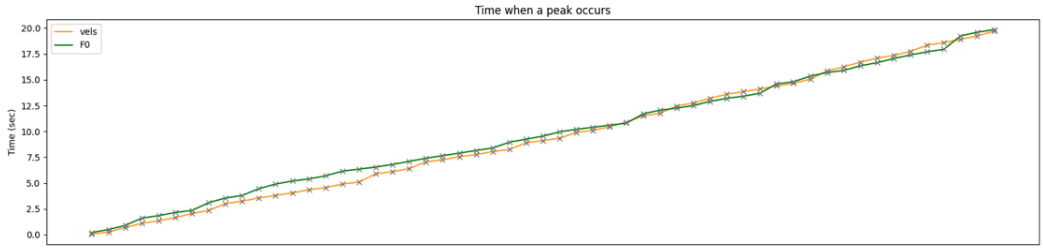

Figure 6.3: Peaks delay for correctly associated pairs.

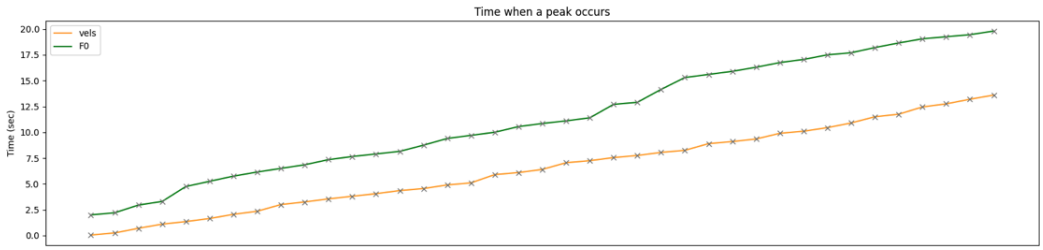

Figure 6.4: Peaks delay for wrongly associated pairs.

For example, the velocity at time $t-1$ cannot be significantly different at time t, just as the position of the hand cannot be near the chin at time $t-1$ and near the hip at time t. Ignoring this sequential frame-by-frame influence leads to the generation of non-smooth gestures for S2G models, and to incorrect results when performing the evaluation.
In a previous work on gesture generation [28], they found that enriching each frame with information about the previous 30 frames and the following 30 frames improve the quality of generated gestures. In this work, then, is made use of context.
A visual demonstration is shown in Figure 6.5.

6.3 Model Architecture

In Chapter 4 were presented state-of-the-art S2G models with the objective of predicting gestures in accordance to the inputted speech. From another perspective, what these models learn is to produce gestures correlated with the speech and, then, the audio-gesture correlation. Therefore, in order to find the most performing model architecture, it was decided to

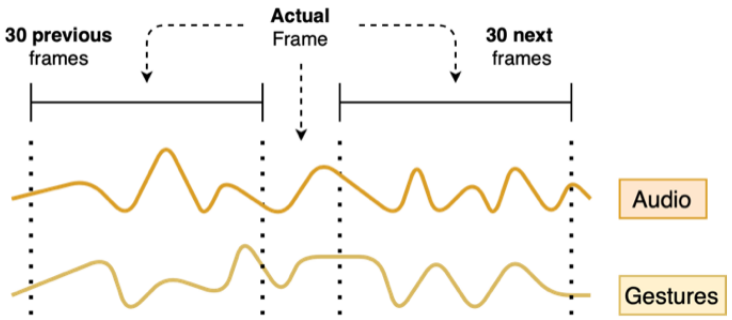

Figure 6.5: A demonstration of the use of *context*.

build it starting from state-of-the-art S2G models, and then work on them by adding, removing or editing some structural parameters to best fit model requirements presented in Section 5.2.

The model architecture to start with is taken from the work *"Analyzing Input and Output Representations for Speech-Driven Gesture Generation"* by T. Kucherenko [28] described in Chapter 4. The architecture used in this work is a time-distributed GRU, that is reusable for the purpose of learning the correlation between inputted audio and predicted gestures. However, the exactly same architecture didn't fit this objective. Indeed, the model had difficulties in generalise its knowledge and tended to perform very well on training data but dramatically on test data, i.e., the model overfitted. The overfitting can be due to different causes, the first thing to do in this case is to try with a simpler architecture with fewer parameters.
There were many trials in which the main structure (GRU) remained untouched but the number of layers and artificial neurones in it decreased. After many trials, the result is a very simply architecture with a GRU wrapped into two fully connected layers.
A visual representation of the final model architecture is shown in Figure 6.6, while Table 6.3 presents the performance of most relevant tested models.

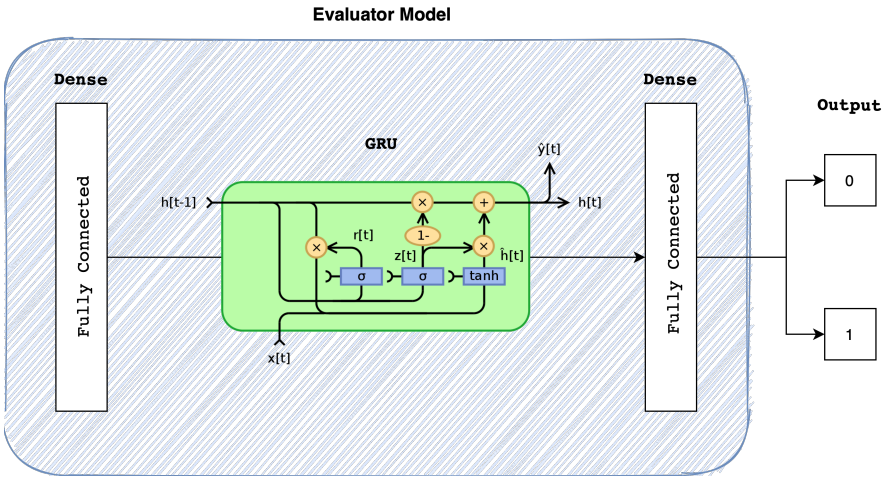

Figure 6.6: Model architecture. GRU image taken by [2]

Model	BS	LR	LF	Loss	AccuracyTR	AccuracyTE
LSTM	2056	0.01	BCE	0.413	62%	58%
LSTM	1028	0.01	BCE	0.223	71%	71%
GRU	**2056**	**0.001**	**BCE**	**0.179**	**92%**	**90%**
GRU	2056	0.01	BCE	0.232	83%	65%

Table 6.3: Performance of tested models. In bold the final model.
BS = Batch Size; LR = Learning Rate; LF = LossFunction; TR = TrainingSet; TE = TestSet; BCE = Binary Cross Entropy.

6.4 Pipeline

A visual representation of the overall pipeline is shown in Figure 6.7. It can be summarised in these steps:

1. Build a new dataset: starting from a speech-to-gestures dataset, is built a new labelled dataset to be used in a supervised fashion.

2. Feature extraction: most relevant features are selected and extracted to retrieve relevant information from data.

3. Prepare input data: create ready-to-use vectors for the evaluator model. Each frame contains information of 30 previous frames and next 30 frames.

4. Train: the model is trained in a supervised fashion.

5. Input: takes a pair <*Audio, Gestures*> as input. Features used are F0 for audio and velocity for gestures.

6. Output: the probability distribution expressing how likely audio and gestures features are correlated.

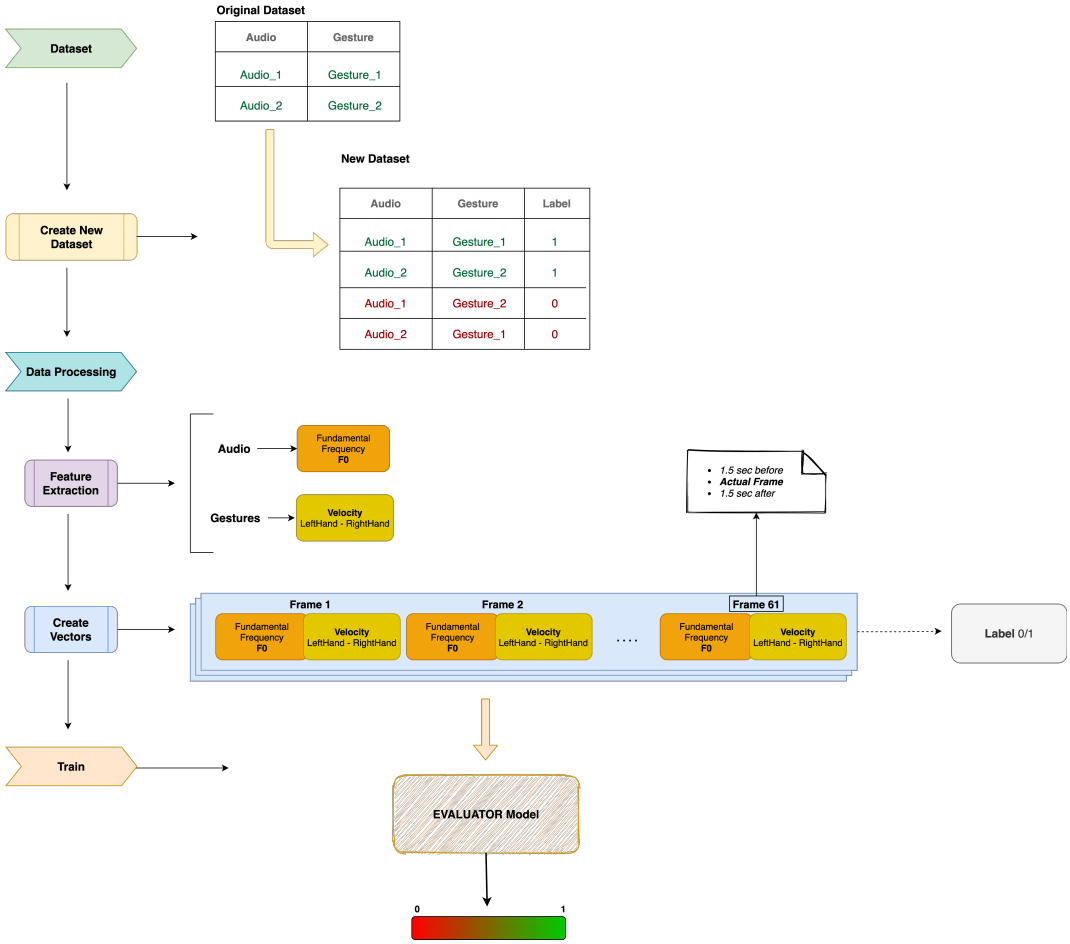

Figure 6.7: Overview of the project pipeline.

Chapter 7

Development and Training

In this Chapter it is described the technical implementation of the main steps presented in the design phase (Chapter 6). In particular, are presented technical details and code snippets about the building of the new dataset in Section 7.1.1, the extraction of features with the use of context and the creation of ready-to-use data vectors in Section 7.1.2, and the model training in Section 7.2.

7.1 Implementation

7.1.1 Dataset

In order to create the new labelled dataset to be used for the training and testing of the binary classification model, it is needed to manipulate the original speech-gestures dataset by GENEA2020 [3].

From a technical perspective, the algorithm is:

1. From the original dataset, label each entry as 1, i.e. "Correlated".
2. From the original dataset, take all the audio-gestures pairs and shuffle their order..
3. Combine them to create new pairs.
4. Label new pairs as 0, i.e. "Not Correlated".

The complete code for this algorithm is shown in Listing 7.1.

Listing 7.1: Build A New Dataset

```
"""
This script create a labelled dataset.
It creates len(OriginalDataset)*2 entries <audio, gesture, label>.
The new labelled dataset is saved as a CSV file.

author: @Famosi
"""

import os
import random
import pandas as pd
import numpy as np


def correct_pairing(speech_dir, motion_dir):
    cp = []
    audios = os.listdir(speech_dir)
    motions = os.listdir(motion_dir)

    assert len(audios) == len(motions)

    n_pair = len(audios)

    # for each motion file, find the related audio file and label it
        as "1"
    for idx in range(n_pair):
        motion = motions[idx]
        picked = motion.split('.')[0]
        audio = [audio for audio in audios if audio.split('.')[0] ==
            picked]
        data = [f'{speech_dir}/{audio[0]}',
            f'{motion_dir}/{motion}', 1]
        cp.append(data)

    return cp


def wrong_pairing(speech_dir, motion_dir):
    wp = []

    audios = os.listdir(speech_dir)
    motions = os.listdir(motion_dir)
```

```
    assert len(audios) == len(motions)

    n_pair = len(audios)

    for _ in range(n_pair):
        # Search for a wrongly associated pair
        while True:
            sampling = random.sample([i for i in range(0, n_pair)], 2)
            if audios[sampling[0]].split('.')[0] !=
                motions[sampling[1]].split('.')[0]:
                break

        # Save pair and label as "0"
        data = [f'{speech_dir}/{audios[sampling[0]]}',
            f'{motion_dir}/{motions[sampling[1]]}', 0]
        wp.append(data)

    return wp


def pair_n_label(speech_dir, motion_dir, csv_file):
    # Pairing
    cp = correct_pairing(speech_dir, motion_dir)
    wp = wrong_pairing(speech_dir, motion_dir)
    print(f'   |- Done!')

    pairs = np.array(cp + wp).T
    d = {'audio': pairs[0], 'gesture': pairs[1], 'label': pairs[2]}

    # Save DataFrame as CSV
    from pathlib import Path
    Path(csv_file.split('/')[0]).mkdir(parents=True, exist_ok=True)
    pairs_df = pd.DataFrame(data=d)
    pairs_df.to_csv(csv_file, index=False)

    print(f'   |- You can find the CSV file at \'{csv_file}\'')
```

The output is a new, labelled, dataset with a number of entries labelled "Correlated" equal to the number of entries labelled "Not Correlated".

7.1.2 Feature Extraction

The feature extraction phase consists in manipulate available data in order to retrieve interesting information (features) and prepare data as input for the final model. As described in Chapter 6, features to be extracted are:

- Fundamental frequency (F0) for audios.
- Mean Hands Velocity for gestures.

Audio - F0 Extraction

The F0 extraction from an audio in *wav* format is possible using *PyReaper*, the python library described in section 5.3. This library allows to extract F0 directly from a *wav* file using the codes shown in Listing 7.2.

Listing 7.2: The use of PyReaper.

```
import pyreaper

'''
x = input audio signal
fs = sampling frequency
'''
f0 = pyreaper.reaper(x, fs)
```

Gestures - Hands Velocity

As for gestures, the need of computing hands velocity require a more complex work. Motion files available in the dataset are in BVH format, containing, for each frame, the rotation angle of each joint. In order to extract velocity from this data, first it is needed a conversion from joints rotation angles into joints 3D positions. Then, after selecting LeftHand and RightHand joints positions, the velocity for each joint can be computed by calculating the first derivative of the joints position for each frame. Finally, the mean of the left and right velocity, for each frame, is computed and used as input feature. In summary, the steps are:

1. BVH to 3D Coordinates: by using *bvhtoolbox* library (see Chapter 5) it is possible to convert a BVH motion file in a CSV file describing for each joint, and for each frame, the position in a 3D space.

2. Select only LeftHand and RightHand joints 3D positions.

3. From 3D hands coordinates to Velocity: by computing the first derivative of positions.

4. Compute, for each frame, the mean between LeftHand and RightHand velocity using the *numpy* python library.

These steps are performed for all <*audio, gestures*> pairs available in the dataset.

Prepare Data for Training

Having extracted F0 frequency and Velocity features, it is possible to prepare data for training. In particular, what is needed is that each audio and gesture frame contains not only F0 and Velocity information for that frame, but also from previous and following *context*. In accordance with requirements describe in Chapter 7, each frame must contains features for 30 previous frames and 30 following frames.

7.2 Training

Set	**Audio-Gestures (input X)**	**Label (target Y)**
Train	(260181, 61, 2)	(260181,)
Validation	(123896, 61, 2	(123896,)
Test	(28909, 61, 2)	(28909,)

Table 7.1: Shape of Train, Validation and Test set.

The dataset contains <audio, gestures, label> entries for a total of 10 hours speech sequences. It was divided into training set, validation set and test set for 70%, 20% and 10% respectively. Final vectors shapes are presented in Table 7.1. A visual demonstration of the training set is shown in Figure 7.1.

The model architecture was trained for 50 epochs with a learning rate (LR) of 0.001 and a batch size (BS) of 2560 samples. As loss function (LF), the binary cross-entropy loss (BCE) was used.

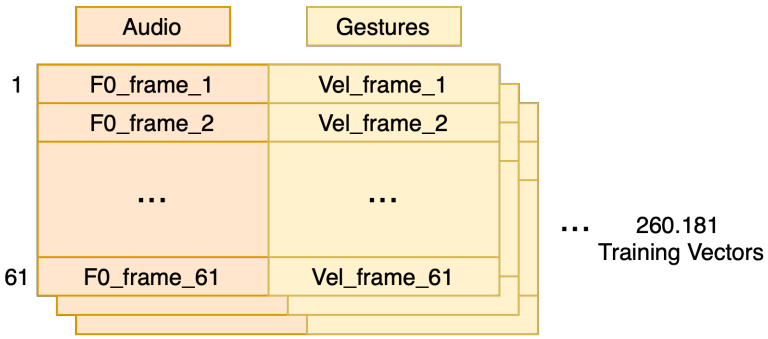

Figure 7.1: A demonstration of the training set shape.

Chapter 8

Evaluation and Applications

The evaluator model is a binary classification model trained in a supervised fashion. For such a model, common used evaluation metrics are accuracy, loss, confusion matrix and F1 score.

8.1 Evaluation Metrics

Accuracy measure the performance of the model. It's defined as:

$$\text{Accuracy} = \frac{\text{No of correct predictions}}{\text{Total no of predictions}} \tag{8.1}$$

Loss is defined as the difference between the predicted value by the model and the true value. The common used loss function for binary classification model is the *binary cross-entropy*, defined as:

$$\text{Binary cross-entropy} = -\sum_{i=1}^{n}\sum_{j=1}^{2} y_{i,j} log(p_{i,j}) \tag{8.2}$$

where, $y_{i,j}$ denotes the true value i.e. 1 if sample i belongs to class j and 0 otherwise, and $p_{i,j}$ denotes the probability predicted by the model of sample i belonging to class j.

Confusion Matrix is a matrix that shows the number of true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN).

Confusion matrix provides also other two interesting values to be calculated from:

- **Recall**: is defined as $\frac{TP}{P}$.
- **Precision**: is defined as $\frac{TP}{P^*}$.

where $P = TP + FP$ and P^* = TP + FN.

F1-score is defined as the harmonic mean between precision and recall. It is used as a statistical measure to rate performance. F1-score is a value between 0 and 1; 0 being lowest and 1 indicating perfect precision and recall, and is defined as:

$$\text{F1-score} = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}} \tag{8.3}$$

8.2 Results

Accuracy and loss results are shown in the training history plot in Figure 8.2. The history plot shows that after 50 epochs, the model reached an accuracy of 91% and a low loss around 0.175.
As for the confusion matrix (see Figure 8.1), the number of TN and FN are similar, meaning that the model is balanced and has no preference in saying a pair is "Correlated" or "Not Correlated". F1-score value is 0.941, meaning that the model has both an high precision and a high recall.

8.3 Applications

In this section are presented two application use cases for the evaluator model. Both of them are based on the replication of a subjective evaluation study from state-of-the-art works in S2G models. The idea is to replicate those user studies in order to:

- Show an application use case: evaluate state-of-the-art S2G models.
- Evaluate the proposed model itself: after the replication of a user study, it might be interesting to check whether the evaluation performed by the model fits a human evaluation.

Confusion Matrix

	Actual 0 - Bad	Actual 1 - Good
Predicted 0 - Bad	TN - 10.194	FN - 1.092
Predicted 1- Good	FP - 936	TP - 16.687

Figure 8.1: Confusion matrix on test set.

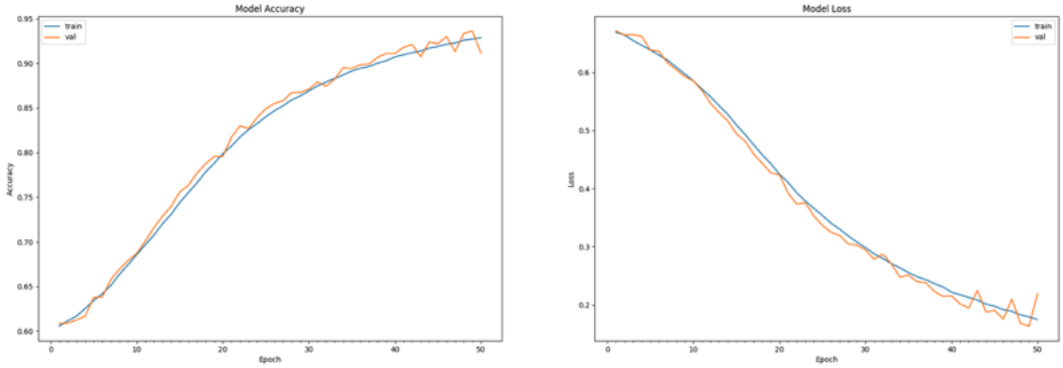

Figure 8.2: Training history for *Accuracy* and *Loss* over 50 epochs.

The first replication aims to evaluate *Gesticulator*, the S2G model proposed by T. Kucherenko [31] replicating an ablation study, while the second one has the objective to evaluate and compare state-of-the-art S2G models replicating the subjective evaluation from GENEA2020 workshop [3].

8.3.1 Evaluation of a S2G Model

In *Gesticulator* by T. Kucherenko, the S2G model is evaluated using statistics measures for the objective evaluation, and performing an ablation study for the subjective evaluation. In the ablation study, human raters were asked to give a score on the preference of the proposed model - called

FullModel - over the following sub-models derived from it:

- NoAutoregression: the *FullModel* without performing autoregression.
- NoPCA: the *FullModel* without performing principal component analysis (PCA) for feature extraction.
- NoFilm: removing their proposed *Film* layer.
- NoText: without the use of text as input feature for the model.

Setting

Participants, recruited on Amazon Mechanical Turk (AMT), were assigned to one specific comparison of two systems. Each participant was asked to evaluate 20 speech video pairs on four subjective measures: "In which video..."

- (Q1) "...are the character's movements most human-like?"
- (Q2) "...do the character's movements most reflect what the character says?"
- (Q3) "...do the character's movements most help to understand what the character says?"
- (Q4) "...are the character's voice and movement more in sync?"

For the replication experiment, results from the Q4 are taken into account, as it is a question that is similar to what the evaluator model should have learned and should predicts.
The replication of this experiment consisted in using the evaluator model in place of human raters.

Implementation

In order to replicate the original setting, the evaluator model was used to rate the FullModel and all the sub-models on each of the 20 videos. The overall score for each model is calculated as the average of all the individual scores for each video. Finally, the preference over the FullModel is calculated by subtracting each sub-model score to the FullModel one.
The code is shown in Listing 9.1.

Note that in this study the Evaluator model gives a score between 0 and 1 for each video. Human raters, instead, were asked to decide, between two videos, which one performed better in a side-by-side comparison. Therefore, in this study it is not interesting that the numerical values of the two evaluations correspond, but rather whether the overall result (e.g., model X is better than model Y) matches.

Listing 8.1: Preference over FullModel.

```
import numpy as np

"""
Input:
models = ["FullModel", "NoPCA", ...]
videos = [video_1, video_2, ..., video_20]
"""
def getPreferenceOverFullModel(model_types, videos):
   model_rate = dict()
      for model in model_types:
         rates = []
         # evaluate() returns the mean rate over all videos for a
             specified model
         mean_rate = evaluate(model_types, videos)
         model_rate[model_types] = mean_rate

      preference_over_fullmodel = [model_rate['FullModel'] -
          model_rate[model] for model in model_types if model !=
          "FullModel"]

      return preference_over_fullmodel
```

Results

Results show that the evaluator model predictions are consistent with the original user study. Indeed, both the Evaluator and human raters state that:

- "NoPCA" and "NoFilm" are preferred over the "FullModel".
- "FullModel" is preferred over "NoText" and "NoAutoregression" models.

A visual representation of these results is shown in Figure 8.3.

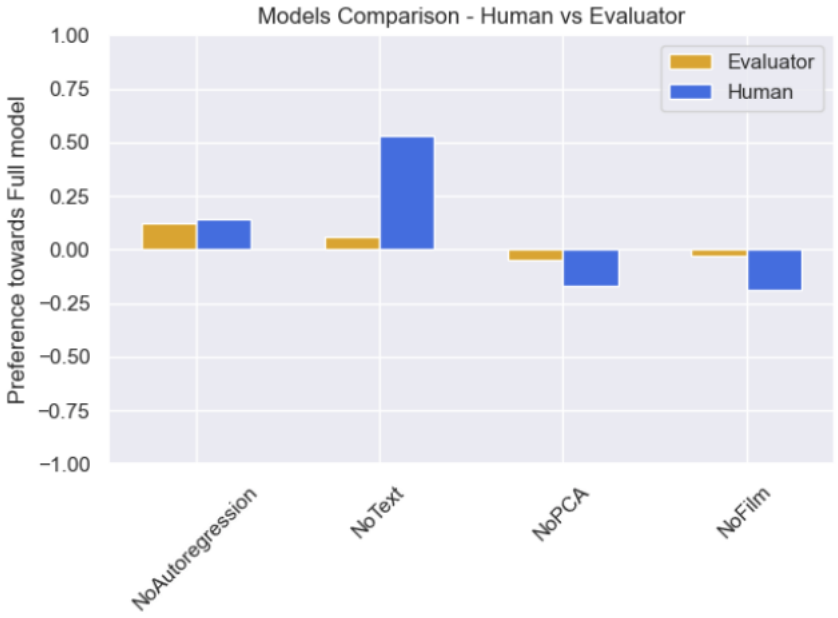

Figure 8.3: Comparison between replication and original evaluation.

8.3.2 Comparison between SoTA S2G Models

One of the most interesting aspects of the GENEA2020 workshop (presented in Chapter 4) is that it provides researchers with a common dataset to work with. This not only provides a reference point for the implementation and training of S2G models, but also a benchmark for the evaluation and comparison of the works presented by the teams involved.
Indeed, at the end of the "call for papers" phase, each model was evaluated performing a *"large-scale, crowd sourced, joint, and parallel evaluation of the motion submitted by the participating teams"* [3].

Setting

Participants for the subjective evaluation were recruited from English-speaking countries through the Prolific Academic crowd sourcing platform. Each participant was asked to rate 40 videos, with an average duration of 10 seconds, with a score from 0 to 100, without the evaluating user knowing which of the submitted models had generated those gestures. In two different studies, two different aspects were asked to be investigated:

- Human-likeness: "How human-like does the gesture motion appear?".

This study aimed in measuring the quality of generated gestures in general.

- Appropriateness: "How appropriate are the gestures for the speech?". This question aimed to investigate the perceived link between gestures and audio in terms of rhythm and timing.

Again, the appropriateness study has the aim to measure what the Evaluator model, proposed in this work, should have learned. Thus, it was taken into account for the replication. In Figure 8.4 are listed gestures and teams that participated in the original evaluation. In this work a subset of them was taken into account.
Natural (N) gestures are GT gestures. In the original study the evaluation of real, natural, gestures allowed to have a metric reference for human evaluation scores, while in the experiment replication, it was useful as an evaluation for the model itself.

			Inputs used?		Representation or features		Stochastic
Name or description	Origin	ID	Aud.	Text	Input speech	Output motion	output?
Natural motion	-	N	✓	✓	–	–	✓
Mismatched motion	-	M	✗	✗	–	–	✓
Audio-based baseline	Kucherenko et al. [15]	BA	✓	✗	MFCC	Exp. map	✗
Text-based baseline	Yoon et al. [27]	BT	✗	✓	FastText†	Rot. matrix	✗
AlltheSmooth	CSTR lab, UEDIN, Scotland	S...	✓	✗	MFCC	Joint pos.	✗
Edinburgh CVGU	CVGU lab, UEDIN, Scotland	S...	✓	✓	BERT† and mel-spectrogram	Rot. matrix	✓
FineMotion	ABBYY lab, MIPT, Russia	S...	✓	✓	GloVe† and mel-spectrogram	Exp. map	✗
Nectec	HCCR unit, NECTEC, Thailand	S...	✓	✓	Phoneme, Spacy word vectors†, and audio features	Exp. map	✗
StyleGestures	TMH division, KTH, Sweden	S...	✓	✗	Mel-spectrogram	Exp. map	✓

Figure 8.4: Table from the GENEA2020 paper [3]. "Conditions participating in the evaluation. Teams are sorted alphabetically by name. The anonymised IDs of submitted entries begin with the letter 'S' followed by a second, randomly-assigned letter in the range A through E, but which letter is associated which each team is not revealed in order to preserve anonymity."

Implementation

```
# Returns a dictionary with scores for each model.
def replicateGENEA(model_types):
    """
    models_rate = {
```

```
    "N": score_N
    "SE": score_SE
    ...
}
"""
models_rate = dict()

for model in model_types:
    # Load Audio and Gestures predicted by the current model
    data = loadData(model)
    models_rate[model] = evaluate(data)

return model_rate
```

Results

The histogram in figure 8.5 shows results from the replication experiment.

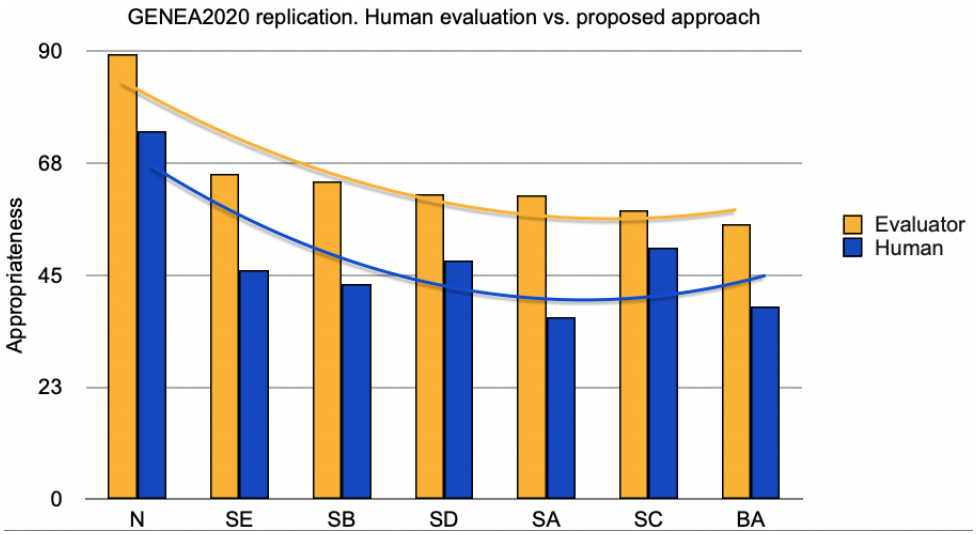

Figure 8.5: Results from the replication of the GENEA subjective evaluation.

Natural (N) gestures received an high score both from the Evaluator and human raters. As for the submitted models, predictions do not completely fit the evaluation by human raters. The trend lines (see Figure 8.5), though, are similar, meaning that there is a certain correspondence between the two evaluations.

Chapter 9

Conclusion and Future Works

Conclusion

In this study it was proposed a novel approach for the evaluation of gestures generative models by providing Evaluator, a ML model that is able to give an overall assessment on the quality of generated gestures. In particular, the presented model provides an evaluation metrics that state how good is the correlation between the audio F0 frequency and the velocity of the hands in a certain speech.
Such a measure is useful to overcome two main problems in speech-driven gesture generation:

1. Statistic measures for the evaluation of S2G state-of-the-art works are dependent to GT gestures, not enabling an objective evaluation of gestures produced by a synthetic voice (e.g., robotics voice). The use of a measure that do not makes use of GT gestures, allows for an audio-source independent evaluation.

2. Each research team works in its own environment and using its own dataset, leading to a limitations in the objective comparison of state-of-the-art models. The presented model can be used as a benchmark for the evaluation of gesture-generating systems.

Results from this study are interesting: the model achieved an high accuracy and the replication of subjective evaluations from recent state-of-the-art works provided an additional measure to asses that the model is reliable. Referring to the result from the replication of the GENEA2020 subjective evaluations (see Figure 8.5), the discrepancy in the rating of submitted models may lead to the following consideration:

state-of-the-art S2G models and, in general, behaviours generation models, are developed to be integrated into ECAs, whether physical or virtual, with the intention to be used by humans and interact with humans. Thus, it is fair to think that human raters have the "first word" when evaluating whether gestures performed by an ECA are, e.g., in accordance with the speech. On the other hand, results from the GENEA2020 user studies highlighted an odd aspect in this sense. In GENEA2020 subjective evaluation study, users were not only asked to rate GT gestures (N) and ones produced by submitted S2G models, but were also asked to evaluate "mismatched" (M) gestures in which audio and natural (N) gestures were not correctly aligned, resulting in wrongly associated pairs. Human raters participating in this study rated the M gestures as better than any submitted model during the workshop (see Figure 9.1). Although the following observation deserves further study and specific research to be confirmed, it seems that humans tend to be influenced by the naturalness of the gestures, since users always preferred natural human-like gestures even in case of out of sync gestures (mismatched - M). This may lead to unreliable user studies and measurements when raters are requested to focus only on a specific characteristic of produced gestures, e.g. on their acoustic-link, and not on their human-likeness. In this case an "agnostic" AI model can be useful to overecome this limitation in subjective evaluations.

However, the proposed approach has its own limitations, mostly due to the fact that this study is unique of its kind. The lack of similar studies in literature has led difficulties in researching which features best represent the relationship between audio and gestures, as well as on the best model architecture to be used for training.
The main limitation of the proposed model is that it focuses only on hand speed and its variation in relation to the frequency F0, without considering other specifications, such as hands positions. This is a critical aspect in the evaluation of generated gestures, leading to a very specific and limited measurement. It is likely that the Evaluator will give a high score to gestures that have a good synchronisation with the input audio while being very different from the expected gestures, for example in their shape and position. For example, imagining a perfect audio-gesture synchronisation in which the hands are always positioned behind the back or under the legs, the Evaluator would give a high score.

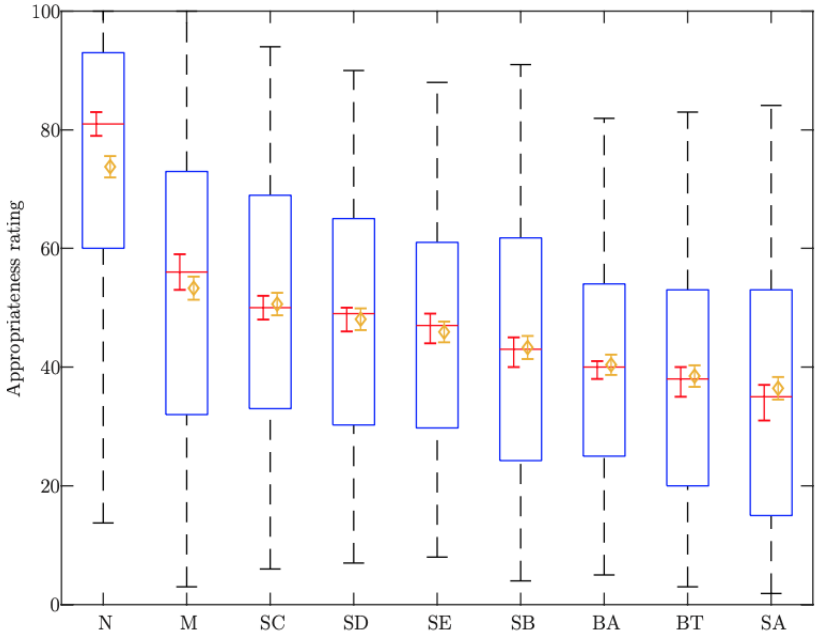

Figure 9.1: Subjective evaluation from GENEA2020 workshop. Mismatched (M) gestures were evaluated by human raters as better than any other submitted model. Image by [3].

Future Works

In future works, an in-depth features study may avoid focusing exclusively on the correspondence between F0 frequency and hands speed, allowing for more complex and exhaustive evaluations to be carried out. Training on a larger number of features not only provides a plausible solution for the "feature limitation" described above, but will also enable other researcher to reuse the proposed model for other purposes. For example, a researcher handling with a robot that is only capable to move its head, might be interested in the relationship between audio and head movements, and should use different features for training. This work provides not only a pre-trained model but also an architecture that can be trained by other researchers using different features, depending on the goal to be achieved and the purpose of the training.

Another aspect to be considered is that the dataset on which the Evaluator model was trained contains speeches made by a single male speaker. This may not be a problem itself, but each person may gesture differently. Using a dataset that contains audio and motion files for several speakers may improve the generalisation capabilities of the model.

As the Evaluator model learns to discriminate between good and bad gestures in relation to the speech, it might be interesting to directly use this model when learning to generate gestures. For example, a speech-to-gesture model architecture can be augmented by adding the Evaluator model as a Discriminator in a GAN setting or as a reward function in a reinforcement learning one.

Ringraziamenti

Un ringraziamento speciale lo devo ai miei genitori. Hanno creduto in me in un momento in cui neanche io credevo in me stesso, dandomi la possibilità e la forza di affrontare questo percorso. Ringrazio tutti i miei familiari e gli amici più stretti che, anche se distanti, mi sono stati sempre vicini. Ringrazio i miei coinquilini e gli amici dell'università, durante questi anni siamo cresciuti aiutandoci l'uno con l'altro, abbiamo lottato, festeggiato e passato momenti indimenticabili. Ringrazio Paulette, che in un anno difficile come il 2020, mi ha supportato e accompangato stimolandomi a fare e dare sempre di più; non so come avrei fatto senza di te.
Ringrazio nonno Romano, mi ha accompagnato in ogni ostacolo che ho incontrato e ogni obbiettivo che ho raggiunto durante questo percorso. Posso solo immaginare quanto sarebbe fiero del percorso che ho intrapreso e dei risultati che ho ottenuto. Questa tesi è dedicata a lui.

Bibliography

[1] Recurrent Neural Network - RNN. Recurrent neural network — nerd-coder., 2020. [Online; accessed 26-February-2021].

[2] Gated Recurrent Unit. Gated recurrent unit — Wikipedia, the free encyclopedia, 2020. [Online; accessed 25-February-2021].

[3] Taras Kucherenko, Patrik Jonell, Youngwoo Yoon, Pieter Wolfert, and Gustav Eje Henter. The genea challenge 2020: Benchmarking gesture-generation systems on common data. 2020.

[4] Communication. *The Oxford English Dictionary*. Oxford University Press, 2020.

[5] Pierre Feyereisen and Jacques-Dominique De Lannoy. *Gestures and speech: Psychological investigations*. Cambridge University Press, 1991.

[6] Sharice Clough and Melissa C Duff. The role of gesture in communication and cognition: implications for understanding and treating neurogenic communication disorders. *Frontiers in Human Neuroscience*, 14, 2020.

[7] David McNeill. *Hand and mind: What gestures reveal about thought*. University of Chicago press, 1992.

[8] Jana M Iverson and Susan Goldin-Meadow. Gesture paves the way for language development. *Psychological science*, 16(5):367–371, 2005.

[9] Laura Valenzeno, Martha W Alibali, and Roberta Klatzky. $\text{Teachers}\surd gesturesfacilitatestudents\surd learning : Alessoninsymmetry. ContemporaryEducationalPsychology, 28(2) : 187 - -204, 2003.$

[10] Melissa A Singer and Susan Goldin-Meadow. Children learn when their teacher's gestures and speech differ. *Psychological Science*, 16(2):85–89, 2005.

[11] Susan Wagner Cook, Howard S Friedman, Katherine A Duggan, Jian Cui, and Voicu Popescu. Hand gesture and mathematics learning: lessons from an avatar. *Cognitive science*, 41(2):518–535, 2017.

[12] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[13] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[14] Marcus Thiebaux, Stacy Marsella, Andrew N Marshall, and Marcelo Kallmann. Smartbody: Behavior realization for embodied conversational agents. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 1*, pages 151–158, 2008.

[15] Hiroki Tanaka, Hideki Negoro, Hidemi Iwasaka, and Satoshi Nakamura. Embodied conversational agents for multimodal automated social skills training in people with autism spectrum disorders. *PloS one*, 12(8):e0182151, 2017.

[16] Michael A Goodrich and Alan C Schultz. *Human-robot interaction: a survey*. Now Publishers Inc, 2008.

[17] Masahiro Mori. The uncanny valley: The original essay by masahiro mori. *IEEE Robots &*, 2017.

[18] WonJoon Chung and Cliff Sungsoo Shin. *Advances in Affective and Pleasurable Design: Proceedings of the AHFE 2017 International Conference on Affective and Pleasurable Design, July 17–21, 2017, The Westin Bonaventure Hotel, Los Angeles, California, USA*, volume 585. Springer, 2017.

[19] Bert Baumgaertner and Astrid Weiss. Do emotions matter in the ethics of human–robot interaction? artificial empathy and companion robots.

In *International symposium on new frontiers in human–robot interaction, London, UK*, 2014.

[20] Jeremy N Bailenson, Nick Yee, Dan Merget, and Ralph Schroeder. The effect of behavioral realism and form realism of real-time avatar faces on verbal disclosure, nonverbal disclosure, emotion recognition, and copresence in dyadic interaction. *Presence: Teleoperators and Virtual Environments*, 15(4):359–372, 2006.

[21] Wim Pouw, Steven J Harrison, and James A Dixon. Gesture–speech physics: The biomechanical basis for the emergence of gesture–speech synchrony. *Journal of Experimental Psychology: General*, 149(2):391, 2020.

[22] Youngwoo Yoon, Woo-Ri Ko, Minsu Jang, Jaeyeon Lee, Jaehong Kim, and Geehyuk Lee. Robots learn social skills: End-to-end learning of co-speech gesture generation for humanoid robots. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 4303–4309. IEEE, 2019.

[23] Ted talks. `https://www.ted.com/talks`.

[24] Carlos T Ishi, Daichi Machiyashiki, Ryusuke Mikata, and Hiroshi Ishiguro. A speech-driven hand gesture generation method and evaluation in android robots. *IEEE Robotics and Automation Letters*, 3(4):3757–3764, 2018.

[25] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[26] Dai Hasegawa, Naoshi Kaneko, Shinichi Shirakawa, Hiroshi Sakuta, and Kazuhiko Sumi. Evaluation of speech-to-gesture generation using bi-directional lstm network. In *Proceedings of the 18th International Conference on Intelligent Virtual Agents*, pages 79–86, 2018.

[27] Taras Kucherenko, Dai Hasegawa, Naoshi Kaneko, Gustav Eje Henter, and Hedvig Kjellström. On the importance of representations for speech-driven gesture generation. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 2072–2074, 2019.

[28] Taras Kucherenko, Dai Hasegawa, Gustav Eje Henter, Naoshi Kaneko, and Hedvig Kjellström. Analyzing input and output representations for speech-driven gesture generation. In *Proceedings of the 19th ACM International Conference on Intelligent Virtual Agents*, pages 97–104, 2019.

[29] Shiry Ginosar, Amir Bar, Gefen Kohavi, Caroline Chan, Andrew Owens, and Jitendra Malik. Learning individual styles of conversational gesture. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3497–3506, 2019.

[30] Chung-Cheng Chiu, Louis-Philippe Morency, and Stacy Marsella. Predicting co-verbal gestures: a deep and temporal modeling approach. In *International Conference on Intelligent Virtual Agents*, pages 152–166. Springer, 2015.

[31] Taras Kucherenko, Patrik Jonell, Sanne van Waveren, Gustav Eje Henter, Simon Alexanderson, Iolanda Leite, and Hedvig Kjellström. Gesticulator: A framework for semantically-aware speech-driven gesture generation. *arXiv preprint arXiv:2001.09326*, 2020.

[32] Anaconda package manager. `https://anaconda.org/`.

[33] Tensorflow. `https://www.tensorflow.org/`.

[34] Fundamental Frequency - F0. Fundamental frequency — aalto university wiki., 2020. [Online; accessed 26-February-2021].