A Social Robot for Intercultural Pedagogy

Università di Genova

Jerin Joy

DIBRIS - Department of Computer Science, Bioengineering, Robotics and System Engineering

University of Genova

Supervisors:
Prof. Carmine Recchiuto
Prof. Antonio Sgorbissa
Prof. Alessia Bartolini

In partial fulfillment of the requirements for the degree of

Laurea Magistrale in Robotics Engineering

December 19, 2023

Declaration of Originality

I, Jerin Joy, hereby declare that this thesis is my own work and all sources of information and ideas have been acknowledged appropriately. This work has not been submitted for any other degree or academic qualification. I understand that any act of plagiarism, reproduction, or use of the whole or any part of this thesis without proper acknowledgment may result in severe academic penalties.

Acknowledgements

I extend my deepest appreciation to all those who made it possible for me to complete this thesis. My heartfelt gratitude goes to my thesis advisor, Prof. Carmine Recchiuto, for his invaluable guidance and unwavering support throughout the entire thesis process. Prof. Recchiuto not only offered expert academic mentorship but also sparked new ideas that significantly enriched the quality of this work. His prompt responses to my emails and consistent personal support have been instrumental in navigating the challenges of this research. I would also like to express my thanks to Prof. Antonio Sgorbissa and Prof. Alessia Bartolini for their support during the thesis.

My sincere thanks extend to my friends who provided abundant help, offering invaluable assistance, support, and guidance. Their advice, both on research and career matters, has been priceless.

Finally, I must express my profound gratitude to my family for providing unfailing support and continuous encouragement throughout my years of study. This accomplishment would not have been possible without them.

I dedicate this thesis to my family and the scientific community.

Abstract

This thesis explores the use of a social robot, Nao, developed by Softbank Robotics, in facilitating intercultural pedagogy in early childhood education settings. The study is driven by societal shifts and global integration, necessitating inclusive multicultural education. The research involves children from diverse cultural backgrounds engaging with the Nao robot in a dynamic dialogue. The system orchestrates an interactive storytelling experiment, posing culturally relevant questions and generating follow-up questions based on responses in various languages. The technology used in the project includes Python for the client-server communication, Google's text-to-speech and speech-to-text APIs, OpenAI API for translations, and the Protégé ontology database for information retrieval.

A preliminary assessment was conducted through an online questionnaire administered to kindergarten teachers. The feedback provided valuable insights into the practical implications and effectiveness of using social robots in a multicultural educational setting.

The proposed social robot aims to foster collaborative interactions and cultural exchange among diverse children, emphasizing storytelling as a means of language learning. The system navigates language diversity through translation services, providing an inclusive educational tool for promoting global understanding.

In summary, this research contributes to the emerging field of social robots in education, particularly in intercultural pedagogy. By leveraging advanced technologies, the proposed system showcases the potential for positive educational practices that celebrate diversity, laying the foundation for more inclusive and effective early childhood education.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

In light of recent societal shifts, we have observed an inflow of immigrants all over the world. Recent events, such as the Russia-Ukraine war, have significantly impacted global migration patterns. The war has resulted in a massive migration of Ukrainians, with the UNHCR recording roughly 6 million Ukrainian refugees across Europe as of July 2022. Similarly, following the Russian invasion of Ukraine in late February 2022, more than 900,000 Russian citizens and residents are estimated to have left Russia by early October 2022. These large-scale movements of people underscore the urgent need for effective intercultural pedagogy in educational institutions worldwide. As global integration reshapes politics, economies, and societies in the 21st century, it is of utmost importance that multicultural education in daycare centers, nurseries, and preschools be socially inclusive. The major factor that usually prevents this inclusion is the barrier of language between the "local" children and the immigrant children. Despite this, there aren't adequate facilities to facilitate intercultural pedagogy, and the staff are typically not trained for it.

Many previous researches have shown that digital devices like mobilephones, and tablets don't have significant effects on intercultural pedagogy (Konijn *et al.*, 2022). On the other hand, physically embodying social robots like Nao and Pepper has proved their trustiness and friendliness with children in many experiments over the past years (Belpaeme *et al.*, 2018a). The unique features of these social robots, which include the ability to interact with movement, speech, and facial expressions make them ideal for literacy and language learning. Many past studies reviewed by Woo *et al.* (2021) showed that young children enjoyed interacting with social robots during activities like storytelling and singing games. There are many instances in the past where a social robot acted in the role of a tutor or teacher as well (Kennedy *et al.*, 2016).

Recent research trends show an increase in interest in the use of social robots in education (Johal, 2020). The literature shows that as the field matures, setup, methodology, and demographics targeted by social robotics applications appear to stabilize and standardize. The physical aspect of social robots allows them to engage with learners in the real environment, and their increased social presence improves learning results. However, implementing social robots in schools poses significant technical, economic, and logistical challenges. Technical challenges may include ensuring the reliable operation of the robots and integrating them with existing educational technology infrastructure. Economic challenges could involve the high costs of purchasing, maintaining, and updating the robots. Logistical challenges might encompass training staff to use the robots, scheduling their use to maximize benefits, and addressing any safety concerns.

The actual appearance of the robot appears to have minimal influence on the learning outcomes: a study (Belpaeme & Tanaka, 2021)reveals that more human-like robots do not necessarily yield higher learning outcomes, but that the presentation and social presence of the robot are more important for the learning outcomes.

Social robots have been used in translanguaging pedagogy, allowing students to use their entire linguistic repertoire in the classroom (van den Berghe, 2022). This method allows students to include any language they speak at home or in other aspects of their lives. Social robots have a significant edge over other types of technology, such as tablets and computers, in terms of physical and social presence. This could be one of the key reasons why social robots have been proven to be more effective than other types of technology in general in schooling.

In summary, social robots have enormous promise for improving intercultural teaching. Despite the difficulties, these robots have the potential to significantly contribute to language acquisition and cultural exchange among children from diverse backgrounds. This thesis aims to go further on this possibility by conducting an interactive storytelling experiment with a social robot. The purpose is to increase cultural understanding among kindergarten students, contributing to the establishment of more inclusive educational methods.

1.2 Objective

The primary objective of this thesis is to explore the potential of a social robot in facilitating intercultural pedagogy. The experiment utilizes a social robot, Nao, developed by Softbank Robotics. This robot is designed to interact with children in a way that promotes learning and engagement.

The experiment involves presenting an interactive story to kindergarten children. This interactive storytelling session is not a passive experience for the children. Instead, it allows them to interact with the robot, responding to its prompts and questions, and learning simultaneously. This active participation can enhance the children's understanding and retention of the story's content.

The children participating in the experiment hail from various parts of the world, bringing with them a rich tapestry of cultural backgrounds. This diversity is a valuable resource for promoting intercultural understanding. The experiment's main goal is to leverage this diversity to familiarize children with different cultures in a social environment.

During the storytelling session, the robot poses various culturally relevant questions to the children. These questions are designed to provoke thought and discussion about different cultures. The children can respond to the robot in their respective languages, promoting linguistic diversity and inclusivity. The robot, equipped with multilingual capabilities, is capable of understanding these responses and answering them back. Furthermore, it can adapt its line of questioning based on the child's previous reply, creating a dynamic and responsive dialogue.

This thesis proposes the realization of a social robot for intercultural pedagogy, specifically aimed at nurseries and kindergartens. The goal is to facilitate positive, collaborative interactions among culturally and linguistically diverse children by means of storytelling. This approach combines technological innovation with pedagogical principles to create a learning environment that is engaging, inclusive, and culturally enriching.

1.3 Content

This section provides an overview of the thesis layout, highlighting the different chapters and their contents.

Chapter 2: State of the Art

- **Introduction (Section 2.1):** This section sets the stage for the thesis, providing a brief overview of the current state of research in the field.
- **Social Robots in Education (Section 2.2):** Explores the role of social robots in educational settings, covering collaborative learning, early literacy, and the use of robotics as an educational tool.
- **Culturally Sustaining Pedagogy (Section 2.3):** Discusses the concept of culturally sustaining pedagogy, emphasizing the importance of culturally relevant educational practices.

- **Social Robots and Intercultural Pedagogy (Section 2.4):** Investigates the intersection of social robots and intercultural pedagogy, with a focus on supporting unaccompanied migrant children and language learning in primary school children.

- **Large Language Models and Human-Robot Interaction (Section 2.5):** Explores the use of large language models in human-robot interaction, highlighting their role in enhancing communication.

- **Ontology-based Knowledge Representation (Section 2.6):** Examines the use of ontology-based knowledge representation in robotics, providing a structured approach to organizing information.

- **Evaluation of Results (Section 2.7):** Describes the methods used to evaluate the results, including the use of questionnaires such as the Godspeed Questionnaire Series and the Smileyometer.

Chapter 3: Methodology

- **System Architecture Overview (Section 3.1):** Provides an overview of the system architecture, offering insights into the overall design.

- **System Components (Section 3.2):** Details the key components of the system, including the Nao Robot and the software tools such as Protégé, Google Cloud API, and OpenAI API.

- **System Workflow (Section 3.3):** Describes the workflow of the system, from setting up connections and question preparation to text-to-speech conversion, recording, and handling translation requests.

- **Interactive Storytelling (Section 3.4):** Introduces the concept of interactive storytelling, outlining both normal and collaborative story modes.

Chapter 4: Evaluation

- **Evaluation Metrics (Section 4.1):** Discusses the metrics used for the evaluation, providing a framework for assessing the system's performance.

- **Technical Evaluation of Response Times (Section 4.2):** Describes the statistical methods employed to analyze response times, including mean calculation, median determination, and standard deviation analysis.

- **Generated Interaction Assessment (Section 4.3):** Presents the online questionnaire for the evaluation and the results of the evaluation.

Chapter 5: Conclusions

- **Research Review (Section 5.1):** Summarizes the key findings from the research, highlighting the contributions to the field.
- **Limitations (Section 5.2):** Acknowledges the limitations of the study, providing a balanced view of the research scope.
- **Future Work (Section 5.3):** Suggests potential areas for future research and development, building upon the insights gained from the current study.

Chapter 2

State of the Art

2.1 Introduction

The integration of social robots in educational settings has emerged as a promising approach to enhance learning experiences. These interactive robots have the potential to engage learners, provide personalized support, and foster social interactions. This literature review explores the use of social robots in various educational contexts, focusing on collaborative learning, early literacy learning, designing robots as second language tutors, and their impact on culturally and linguistically diverse young children. Through an examination of multiple studies, this review aims to shed light on the effectiveness of social robots in facilitating learning and promoting positive outcomes in educational environments. Understanding the potential benefits, challenges, and design considerations associated with the use of social robots in education is essential for harnessing their full potential and informing future educational practices.

2.2 Social Robots in Education

There is a notable upsurge in interest in the subject of social robotics, especially in educational settings (Mubin *et al.*, 2013). The possibility that social robots would completely transform how humans communicate and learn is what is generating all of this attention. These robots are perfect for creating a more dynamic and engaging learning environment since they can recognize and communicate human emotions (Breazeal, 2002). The ability to recognize and respond to human emotions allows these robots to adapt their behavior and responses according to the learner's needs, thereby creating a more personalized learning experience. A survey (Belpaeme *et al.*, 2018a) conducted by researchers in the field targeted three key concepts: efficacy, embodiment, and interaction role. Some fascinating insights were uncovered by the survey's results. In terms of embodiment, the majority of the experiments were conducted with the Nao robot. This implies that a robot's actual presence can greatly improve the educational process. With its humanoid shape and capacity to mimic human gestures, the Nao robot offers learners a more relatable and captivating interface.

Interestingly, it appears that the primary function of these robots is that of a teacher or tutor (Kennedy *et al.*, 2016). This position is in line with the paradigm change in education from teacher-centered to learner-centered methods (Shimada *et al.*, 2012). Under a learner-centered approach, instructors serve as facilitators rather than directors of learning, allowing students to actively participate in their education. Social robots are a wonderful fit for this paradigm because they encourage active learning and offer tailored guidance and assistance. In addition to their teaching capabilities, these robots are equipped with advanced technologies such as motion sensors, voice recognition, and emotion-sensing software (Cavallo *et al.*, 2018). These characteristics increase the robot's effectiveness in educational environments by allowing it to engage in restricted social interactions with people. For example, emotion-sensing software enables the robot to react properly to the learner's emotional state, while speech recognition allows it to grasp learners' verbal commands.

Commonly used robots like Pepper and its predecessor, NAO, also have a non-threatening appearance that helps with the "uncanny valley" issue (Li *et al.*, 2010). The "uncanny valley" is a phenomenon where humanoid objects that closely resemble humans but are not perfectly human-like can elicit feelings of eeriness or revulsion. Manufacturers have been able to address this problem by giving these robots a nice, non-threatening appearance. As a result, learners are more likely to accept and be taught by these robots. The use of social robots in educational settings has emerged as a viable strategy for improving learning experiences. These engaging, interactive robots have the potential to foster interpersonal relationships, provide tailored support, and engage students. They can design a more lively and engaging learning environment that meets the various demands and learning preferences of the students.

Robotics activities offer a great deal of potential to enhance classroom instruction, according to educational theorists like Papert (1993). According to researchers, the majority of the literature on the application of robotics in education is descriptive in nature and is based on accounts of teachers who used their own initiative to produce successful results (Caci et al., 2003; Petre and Price; 2004; Williams et al., 2007) (Petre & Price, 2004). Another finding from the literature is that, up to this point, the majority of robotic technology applications in education have generally concentrated on assisting the teaching of topics that are closely related to the area of robotics, such as robot programming, robot construction, or mechatronics. In addition, the majority of applications, where the robot was built or programmed, employed it as a means or a passive instrument in the learning activity.

Mitnik *et al.* (2008) claim that "so far, most of the applications of robotic technology in education have focused on supporting the teaching of subjects that are closely related to the Robotics field, such as robot programming, robot construction, or mechatronics." The findings reveal that 80 percent of the research examine physics and mathematics-related themes. However, two studies stand out for having distanced themselves from the area of exact sciences; the first (Whittier & Robinson, 2007) aims to apply robotics to the teaching of fundamental evolutionary principles, and the second (Owens, Granader, Humphrey, and Baron-Cohen, 2008) (Owens *et al.*, 2008) aims to improve social communication skills in autistic people.

Many researchers believe that robotics is a great source of energy that can be used to inspire children's learning when it comes to the usefulness of robotics as a teaching tool. However, Johnson (2003) recalls that before rushing headlong into new education programs, we need to understand exactly what robotics has to offer the educator. In general, the articles' findings indicate that using robotics can increase learning. Contrarily, it should be highlighted that there are instances where using robotics in the classroom has not resulted in a material improvement in student learning, as seen in studies of particular circumstances. According to the research, educational robotics can be used to improve academic performance in particular STEM (Science, Technology, Engineering, and Math) topic areas. We can see that the skills involved in skill development through robotics are concentrated on: (i) thinking skills (observation, estimation, and manipulation); (ii) science process skills/problem-solving approaches (such as solution of evaluation, hypothesis generation, hypothesis testing, and control of variables); and (iii) social interaction/teamwork skills.

The effectiveness of the learner-centered approach over more established teacher-centered alternatives has been shown by learning science. Social robots can be extremely useful in instances involving collaborative learning, in which two or more individuals study or attempt to learn something together. By offering direction, moderating conversations, and providing comments, they can promote collaboration. Exploring a larger range of potential applications has the ability to engage young people with a wider range of interests because the uses of robots in educational settings are unduly limited. This can be accomplished in a number of ways, including through narrative, interdisciplinary integration with the arts and music, etc.

2.2.1 Collaborative Learning With Social Robots

Learning research has shown that the learner-centered approach is more effective than conventional teacher-centered approaches. Collaborative learning, which is when two or more individuals try to learn something together, has drawn a lot of interest in this setting. An intriguing perspective on this strategy is provided by research by Kanda *et al.* (2012). The study used Lego Mindstorms, a well-known tool for teaching robotics and programming principles, to facilitate cooperative learning among a group of sixth graders. In order to help the students, a social robot named Robovie was deployed to run the class and demonstrate how to use the Lego Mindstorms. The study used two different types of Robovie behavior: sociable behavior and non-social behavior. In its social behavior, Robovie roamed the classroom, congratulating students for their good work and offering advice when they made mistakes. This conduct was intended to resemble the social interactions that take place frequently in a classroom setting. Yet, Robovie only engaged in the bare minimum of class management actions in its antisocial conduct.

A pretest and posttest were used to evaluate the efficacy of these actions. Individual ability was determined by the pretest, while student impression of the robot and social acceptance were evaluated by the posttest. The outcomes showed that using a robot in a collaborative learning environment was successful. It's interesting to note that while social behavior was found to positively influence social acceptance, learning success was not significantly impacted. This implies that while social behavior can help students accept robots more readily, it does not necessarily result in better learning outcomes. Social behavior only had an impact in the first two classes, though, which raises the possibility of a "novelty effect." This suggests that the thrill of engaging with a robot can fade with time.

The effectiveness of employing robots to inspire youngsters also seemed to depend on how much of the class participated. Robots were found to be more efficient in classes that were less interesting, indicating that they might be especially helpful in disciplines that are generally regarded as less interesting or demanding. These results lead to the suggestion that a design devoid of social behavior would be more practical for deployment right away. Social behavior, however, might be helpful in encouraging kids to pursue less attractive subjects.

2.2.2 Early Literacy Learning with Social Robots

The area of social robotics has received a lot of interest recently, especially for its use in helping young children learn across a variety of domains. The unique social features of these robots, which include the ability to interact using speech, movement, and facial expressions, make them particularly relevant for literacy and language learning. These robots have the potential to improve children's language and communication skills since they are made to interact with humans in a way that mimics human interaction. The potential of social robots to aid young children in learning literacy and language is discussed in Neumann (2016). These robots' interactive features enable a dynamic learning environment that can change to meet the unique needs of each child. This adaptability is crucial in creating an engaging and effective learning experience that can cater to a wide range of learning styles and abilities.

The research conducted by Neumann (2016) involved developing children ranging in age from 0 to 8 years. The study evaluated 13 studies from different countries, including Israel, Italy, Japan, the Netherlands, the United Kingdom, the United States of America, and Taiwan. The widespread interest in and importance of social robots in early learning are highlighted by this global representation. The research included a variety of humanoid and non-humanoid robots, including NAO, MecWilly, Robovie, RUBI-4, iRobiQ, Dragonbot, and Tega, and they included a wide spectrum of individuals. Through diverse activities like storytelling, games, singing, and language exchanges, these robots interacted with kids. The findings showed that social interaction skills, word reading ability, and vocabulary development in youngsters were all positively influenced by interactive encounters with social robots. Several methodological techniques were used in the experiments, such as experimental controls, comparisons of different robot attributes, and activity conditions.

One of the key theoretical perspectives identified in relation to social robots and language and literacy learning is a robot's capacity to scaffold learning within a child's Zone of Proximal Development (ZPD) (Takayuki Kanda & Ishiguro, 2004). Scaffolding is a process where a more knowledgeable other provides prompts and clues to help a child complete a task (Wood *et al.*, 1976). In one study, the MecWilly robot helped kids by changing the amount of help they received from their ZPD (Vygotsky, 1978). This aligns with the pedagogical viewpoint that child-robot interactions should be tailored to a child's capabilities.

The studies reviewed indicated that young children enjoyed interacting with social robots during activities like storytelling and singing games. Children exhibited positive attitudes toward the robots, viewing them as friends and finding their stories to be both engaging and understandable. Social robots effectively engaged and held children's attention, supporting their vocabulary learning. Integrating social robots in early years classrooms created excitement among students who interacted with the robots, made eye contact, and showed empathy towards them. Building trust and introducing robots in a social manner were important factors influencing children's engagement with the technology. These findings highlight the possible advantages of integrating social robots into young children's language and literacy instruction.

2.2.3 Robotics as an educational tool

In the last decade, robotics has emerged as a significant tool in education, attracting the attention of both teachers and researchers. It has been found to be an effective tool for helping kids in preschool through high school enhance their social and cognitive abilities. It also facilitates learning across a broad spectrum of academic areas, such as physics, math, technology, informatics, and other subjects. The literature review reveals that Educational Robotics is a rapidly growing field with the potential to significantly influence science and technology education at all levels, from kindergarten to university (Alimisis, 2013). Beyond conventional learning paradigms, Educational Robotics has the ability to revolutionize education by providing new opportunities for intellectual advancement. It promotes experiential learning opportunities and student participation. Students gain technical proficiency as well as the ability to think critically, collaborate with others, and solve issues creatively by designing, constructing, and programming robots. These skills are becoming more and more crucial in the digital economy of the twenty-first century. Moreover, educational robotics can improve the rewards and enjoyment of learning. Students' curiosity in STEM subjects might be piqued and their desire to study more can be sparked by the excitement of witnessing their own creation come to life.

Many researchers have endorsed robotics as an effective instructional tool. A significant portion of the literature is dedicated to the use of the Lego MindStorms kit, a versatile tool that has been successfully implemented in educational settings ranging from primary schools to universities (Klassner & Anderson, 2003; Petrovič & Balogh, 2008). Johnson (2002) asserts that the interdisciplinary character of robotics provides special educational benefits. It involves a synthesis of various technical topics, including Mathematics and Physics, Design and Innovation, Electronics, Computer Science and Programming, and Psychology. This multidisciplinary approach promotes a holistic learning experience by giving students a thorough understanding of these subjects.

The pedagogical value of robots is derived from their practical application. According to past studies, a major component of robots' educational value is the process of programming them, which entails applying or expanding knowledge to recognize and solve problems (Bilotta *et al.*, 2009). This hands-on approach not only enhances students' understanding of the underlying concepts but also cultivates their problem-solving skills. Furthermore, because they are physical and complicated, robots are particularly inspiring technological advancements. They provide students with concrete examples of how theoretical concepts can be applied in real-world scenarios. They also satisfy the primal human drive for invention and creativity. The process of designing, building, and programming robots can be a deeply satisfying experience for students, fostering their interest in STEM fields (Bilotta *et al.*, 2009).

Educational robotics is a powerful learning tool that significantly enhances the learning experiences of students through a hands-on, mind-on approach. With this method, students can interact directly with the content, which promotes comprehension and memory retention. Student engagement and motivation are increased by educational robots' engaging and interesting hands-on learning environment (Eguchi, 2014). Educational theorists, such as Papert, have long recognized the immense potential of robotics activities in improving classroom instruction (Papert, 1993). By giving students the chance to apply theoretical information in real-world situations, these activities help to close the knowledge gap between theory and practice. This experiential learning approach not only enhances understanding but also cultivates critical thinking and problem-solving skills.

Furthermore, according to Mikropoulos (2013), robotics can be utilized as MindTools in the classroom. MindTools are cognitive tools that enhance the cognitive powers of learners during problem-solving. In the context of educational robotics, these tools support knowledge construction through the design of meaningful, authentic projects. Through experiential learning in both virtual and real-world environments, this approach fosters active learning and empowers students to take charge of their education. Moreover, educational robotics presents students with cognitive conflicts or challenges that they must overcome. These challenges stimulate cognitive development and foster resilience and perseverance. As students navigate these challenges, they engage in reflective thinking and collaborative problem-solving. These skills are not only essential for academic success but also for personal and professional development.

2.3 Culturally Sustaining Pedagogy

Seventeen years ago, a researcher named Gloria Ladson-Billings (Ladson-Billings, 1995) wrote an important article about a teaching approach called "Culturally Relevant Pedagogy". Her research with effective educators of African-American students served as the foundation for this strategy. The goal of culturally relevant pedagogy is to improve the effectiveness and meaning of learning for all students, irrespective of their language or cultural background. It builds upon the research conducted by numerous scholars who examined language, literacy, and culture between the 1970s and the 1990s. Many studies by the mid-1990s indicated that this strategy could benefit students of color, who were frequently disadvantaged by structural injustices. This was something of a ”golden age” for this kind of study. The purpose of this study was to develop curriculums and new teaching strategies that would shift the emphasis from deficit approaches—which emphasize students' weaknesses—to approaches that acknowledge and build on the strengths that students bring from their varied backgrounds.

In education, "deficit approaches" were prevalent in the 1960s and 1970s. These methods viewed many students' and communities of color's languages, literacies, and cultural practices as issues that required fixing (Paris & Ball, 2009). The intention was to replace these "deficient" practices with the prevalent literacy, language, and cultural norms of the era—many of which were in line with middle-class, White standards. For instance, there were federal "Indian schools" with the goal of displacing Native American languages and customs from Native American communities and students by force. Research on the "culture of poverty" also put forth the idea that impoverished students of color's communities and home cultures were worthless in both society and education. In other words, these deficit approaches aimed to erase the diverse linguistic, literate, and cultural practices that many students of color brought from their homes and communities, and replace them with what were seen as superior practices.

"Culturally Relevant Pedagogy" (CRP) is a teaching approach aimed at making education relevant and responsive to the diverse languages, literacies, and cultural practices of students. They do, however, raise concerns about whether the terms "relevant" and "responsive" adequately preserve and value the various languages and cultures that make up our multiethnic and multilingual society, as well as whether they accurately reflect the teaching and research that they are intended to describe. The words "relevance" and "responsiveness" don't specifically support Ladson-Billings' intention for CRP to support the preservation of cultural practices and critique, Paris (2012) contends. He argues that being "relevant" or "responsive" does not imply that an educational program is dedicated to upholding and appreciating students' heritage while fostering cultural and linguistic diversity. This is in line with the findings of Alim (2007), who concentrated on teaching in Hip Hop cultures. They argue that a new term and methodology are needed to better capture the state-of-the-art in this field of study and practice. This new method ought to assist educators in appreciating and preserving the cultural practices of their students while simultaneously assisting them in acquiring the language, literacy, and other cultural practices that are prevalent in the community.

The author introduces the concept of "Culturally Sustaining Pedagogy" (CSP), an educational approach that they believe encapsulates some of the best practices and research in education. CSP assists students in maintaining their cultural and linguistic competencies while also becoming proficient in dominant cultural practices, in contrast to approaches that are only responsive or relevant to students' cultural experiences. In order to maintain and promote linguistic, literary, and cultural diversity as an integral element of the democratic educational process, CSP aims to cultivate multilingualism and multiculturalism among educators as well as students. In order to preserve a vibrant, diverse society, the author highlights the necessity of both common cultural practices across different groups (like Dominant American English) and unique cultural practices within different groups (like Spanish or African American Language, Navajo or Samoan).

The author also emphasizes how cultural practices are dynamic and ever-changing. They advise against drawing conclusions that associate particular languages and cultural customs with particular racial or ethnic groups. They cite recent studies that cast doubt on long-held beliefs about who "owns" particular languages and cultural customs, demonstrating that young people both preserve and innovate upon traditional forms of their cultures. For instance, African American students express their identities through African American Language (AAL) and Hip Hop cultures, but other young people of color, such as Latina/o and Pacific Islander youth in the U.S., as well as youth across racial and national boundaries globally, also heavily participate in these cultural practices. The author's own research has examined how young people of African American and Pacific Islander descent who live in dynamic urban communities converse in Spanish with their Latina/o peers. Thus, while it is imperative that we preserve the languages and cultures of diverse communities in our pedagogical approaches, we also need to be willing to preserve them in the ways that young people of today live and use them, as well as in their changing forms.

2.4 Social Robots and Intercultural Pedagogy

In the realm of education, the integration of technology has the potential to reshape the way we teach and learn. One fascinating avenue is the use of social robots as educational tools. These advanced machines can offer personalized learning experiences, and their presence can significantly impact the learning environment. To delve into this exciting frontier, we draw insights from two distinct studies that shed light on the design considerations and challenges when using social robots as language tutors, particularly for culturally diverse preschool children.

The first study, "Designing Social Robots as L2 Tutors for Preschool Children" (Belpaeme *et al.* (2018b)), centers around the development of a lesson series aimed at teaching basic vocabulary in a second language to children around the age of five. The key objective is to tailor interactions to each child's unique learning needs while keeping the learning goals within their Zone of Proximal Development (ZPD). In this endeavor, the Nao robot is selected for its ability to produce speech in various languages. However, a notable challenge emerges in the form of automatic speech recognition, which struggles with accurately understanding children's speech. Consequently, the study delves into innovative strategies to overcome these technological limitations.

Research from Moriguchi *et al.* (2011) and Rosenthal-von der Pütten *et al.* (2016) suggests that the mere physical presence of a robot might not be the sole driver of improved language learning outcomes. Nevertheless, there is a wealth of evidence indicating that the robot's physical presence can positively influence various aspects of interactions, including the learning process itself. Thus, the physical presence of the robot remains a central consideration in crafting the learning environment.

Further pedagogical issues surface during the design process of robot tutoring interactions for children. These encompass variations in age, the selection of target words, the creation of meaningful contexts, and the dosage of the intervention. The study identifies five-year-old children as the most responsive learners, highlighting the need for age-appropriate design. Selecting target words is a nuanced process, requiring consideration of semantic coherence, relevance to the content domain, and the child's knowledge of their first language. In terms of intervention dosage, a maximum of six words is recommended for each lesson, with each word being repeated at least ten times throughout the session.

Beyond these factors, the study explores additional essential aspects of child-robot interaction, such as the first interactions, the robot's function, the environment in which the interactions take place, the robot's vocal and nonverbal cues, and the robot's feedback. The researchers created an engaging session that exposes kids to the robot through storytelling, hands-on activities, and behavior observation in order to promote positive engagement and raise kids' comfort levels. By presenting the robot tutor as a peer to the kids, this method blends the appealing features of a robot with adult-style teaching techniques to produce fun and productive language learning experiences.

In the second study, "Culturally and Linguistically Diverse Young Children and Social Robots" (Kim *et al.* (2021)), the focus shifts to the use of social robots in a culturally diverse kindergarten classroom setting. This research stems from the observation that students whose home language is not English tend to perform below their native English-speaking peers in various subjects. To address this educational disparity, a humanoid social robot, known as Skusie, is introduced to an active kindergarten classroom. The study employs the Wizard-of-Oz method to overcome the limitations of automatic speech recognition, allowing a human operator to control the robot's responses in real-time.

A crucial finding of this study relates to anticipating the diverse communication styles of children, further exacerbated by limitations in automatic speech recognition software. Many children use words that approximate their intended meanings rather than precise language. To address this challenge, the researchers employ a human "wizard" to control the robot's actions, ensuring personalized and responsive interactions. Additionally, a "friendly" approach is adopted, with the robot using children's names to enhance communication, leading to more successful interactions.

Designing equitable and culturally sustaining activities for all children proves to be another intricate challenge. The researchers address this issue by designing the robot to be bilingual in Spanish and English, promoting inclusivity among both Spanish and English-speaking students. This bilingual component plays a pivotal role in integrating these two groups of students equally into the learning environment, underscoring the importance of culturally responsive design.

2.4.1 Social Robots supporting the Inclusion of Unaccompanied Migrant Children

A significant number of Unaccompanied Children (UAC) applied for international protection in EU nations in 2016, underscoring the EU's obligation to assist and foster these children's potential to make valuable contributions to society. Even though the number of UAC decreased in 2018, social services play a crucial role in ensuring their protection, access to education, healthcare, and programs that promote their social inclusion. However, it is important to highlight that the situation has changed since then. According to the most recent data from 2023, the number of UACs has fluctuated, with an average of 8,000 to 10,000 children in care each month. This includes children from different countries and ages. These figures have probably been influenced by current global events, such as the conflict between Russia and Ukraine. These recent changes highlight the necessity of social services in assisting UAC and encouraging their social inclusion. A project was created to teach Italian to UAC in order to overcome this difficulty, with an emphasis on gestures associated with culture (Poggi, 2006). Social exclusion is frequently the result of linguistic deprivation, and communication skills are essential for assimilating into a new cultural setting. There are several ways to communicate, and one of the rich and culturally-specific aspects of language is hand gestures. Different gestures have different meanings, and cultural interpretations can differ in how they are understood. Since these gestures are essential to communication, teaching them to students learning a foreign language is important. However, because of their migration journey, UAC may encounter mistrust and fear upon arriving in a new country. A potential remedy is provided by social robots, which are embodied autonomous entities that imitate human social behaviors. Language instruction is one of the many uses for them (Schodde *et al.*, 2017). According to research, kids who learn from robots generally retain new vocabulary better in their long-term memory (Alemi *et al.*, 2015). Social robots are useful tools in therapeutic and intervention settings because they are less complicated and frightening than humans. UAC can be helped to adjust to their new linguistic and cultural surroundings by having them programmed to provide predictable and repeatable interactions.

Unaccompanied Migrant Children, who have the difficult dual status of being minors and migrants, have been flooding Italy in recent years. As a result of their perilous travels and separation from their parents, UAC frequently suffer from trauma and cultural shock. 8,342 UAC, mostly from Albania and North Africa, were being housed in Italy as of March 2019, and their integration required specific assistance. Reception initiatives like BLUE and GREY work to help these kids comprehend Italian language, culture, and customs in order to meet their special needs. Establishing trust and fostering interaction with these traumatized youth, who might object to physical contact, is still very difficult. By offering engaging interactions and language instruction, social robots provide a promising solution to bridge this gap. They aid in connecting cultural mediators and operators with UAC, facilitate the integration process, and help them understand Italian culture and customs. UAC face considerable difficulties when learning Italian due to the language's intricate structure and phonetic distinctions from their native languages. The use of robots is a promising way to promote school learning and friendships. In their first interactions with teachers, UAC may find it difficult to comprehend the tasks and rules, and there may be trust issues related to cultural gender. Personalized assistance and specially designed learning initiatives are necessary to establish a friendly and trustworthy atmosphere. Through the creation of engaging and enjoyable lessons that encourage students to persevere in their learning, robots play a critical role in addressing the main challenge associated with UAC language learning: motivation and attention.

Both verbal and nonverbal components are involved in communication, and one important nonverbal mode of expression is through gestures. Particularly in Italian culture, these hand, arm, and shoulder motions are crucial to communication because they influence rhythm, emphasis, spatial arrangement, and the expression of unspoken desires or thoughts. There are many different kinds of gestures: deictic, which point to things or people; iconic, which show or mimic movements; batonic, which emphasize rhythmic syllables; and symbolic, which have meanings unique to a culture. These gestures are symbolic and socially coded, making them difficult for people from other cultures to understand. For the experiment, researchers utilized the Softbank Nao robot in conjunction with a Microsoft Kinect to detect the gestures exhibited by the child. The robot's gestures were developed using the "Timeline" scripts feature in Choregraphe.

Ultimately, this research delves into the vital responsibility of teaching Unaccompanied Minors (UAC) cultural gestures in order to support their social integration into a foreign cultural environment. Through the integration of Microsoft

Kinect with the Nao Social Robot, the researchers present a novel approach that utilizes the robot's humanoid features to accurately mimic gestures. This approach works well for teaching because it provides for practice and repeated interactions. It is especially helpful for young migrants who might experience trust issues after moving. The study's approach appears to be effective based on preliminary results, despite certain limitations related to the quantity and diversity of subjects. Notably, this study closes a gap in current European initiatives by adding to the emerging field of creative teaching strategies for immigrant populations.

2.4.2 Social Robots for (Second) Language Learning in (Migrant) Primary School Children

The lack of educational professionals continues to be a significant obstacle in the primary education landscape in Europe. This shortfall is especially noticeable in the context of the increasingly diverse and growing student populations brought about by migration. As a result, there is growing interest in utilizing technological advancements like tablets and social robots to close the staffing gap in education. According to the current discourse, tablets have the potential to improve learning outcomes, but because of all the distracting applications they contain, their effectiveness as teacher substitutes is compromised. On the other hand, social robots are thought to promote more natural interactions and help kids learn languages because of their anthropomorphic features and embodiment (Konijn *et al.*, 2022).

The study conducted by Konijn *et al.* (2022) explored the efficacy of social robots in language learning among bilingual or multilingual primary school children in the Netherlands, with a particular focus on those from immigrant backgrounds. The researchers employed a Softbank Nao robot that was fitted with Choregraphe software to involve children one-on-one in three interactive storytelling activities that were intended to improve vocabulary retention. A baseline linguistic ability test was given prior to the exercises, and word retention was assessed through immediate and delayed post-tests. Children who had been familiar with the robot before the experiment were less likely to experience the novelty effects. In this study, the effects on the children's learning outcomes of two robot modalities—one with neutral behavior and the other with sociable behavior—were compared. Additionally, the "Wizard of Oz" technique was employed to circumvent the limitations of Automatic Speech Recognition technology, ensuring precision in the robot's interactions.

The findings of Konijn *et al.* (2022) were multifaceted. Children demonstrated longer-term learning with the robot than with the tablet, according to observations, which highlights the potential of social robots in educational settings. Engagement and enjoyment levels were consistently higher with the robot across all measurement intervals, despite no discernible differences between the learning outcomes of children interacting with the robot displaying social versus neutral behavior. Immediate learning outcomes were significantly predicted by this increased engagement. Furthermore, the study found that regardless of the robot's actions, people thought it looked more human than the tablet. The relationship between the learning outcomes and the device was not mediated by this perceived humanness, though. However, this perceived humanness did not mediate the relationship between the device used and the learning outcomes. Significantly, the study also showed that kids with lower levels of Dutch language proficiency benefited more from the robot interaction, indicating the effectiveness of social robots in assisting students with different language proficiency levels in language learning.

In summary, Konijn et al.'s research from 2022 offers strong evidence that social robots could be more successful as language tutors than tablets at engaging and educating kids, especially when it comes to learning a second language. The field of child-robot interaction in educational settings has greatly benefited from these findings, and they open the door for further research into the long-term effects of using social robots as tutors. The challenges presented by the lack of staff and the growing diversity in primary education in Europe may be greatly mitigated by such technological interventions.

2.5 Large Language Models for Human-Robot Interaction

Large language models (LLMs) such as GPT-3, introduced by Floridi & Chiriatti (2020), represent a breakthrough in the field of natural language processing (NLP), demonstrating an ability to perform language-related tasks with remarkable accuracy. The transformative transformer architecture provides the foundation for these models by enabling them to handle long-range dependencies and contextual relationships within text through its self-attention mechanism. These kinds of architectures have made it easier to create models like BERT (Devlin *et al.*, 2018). These models can be used for a variety of NLP tasks by pre-training and fine-tuning, demonstrating their adaptability and few-shot learning abilities.

Large language models have a number of risks and challenges when used in education, requiring for careful consideration and risk-reduction techniques. Model-generated content that resembles copyrighted material may give rise to copyright issues; therefore, clear permission procedures and strict adherence to copyright requirements are required. Concerns about fairness and bias highlight the significance of varied and representative training data, ongoing observation, and corrective action. It is possible for students and instructors to rely too heavily on models, which can stifle critical thinking and creativity. To counteract this, awareness-raising, the use of models as tools, and the integration of human expertise are some strategies. Since teachers may rely too much on models, it's important to provide them with continual training, value creativity and critical thinking, and keep an eye on their efficient use. The need for research, case-based guidance, training opportunities, and community building arises from educators' lack of understanding and expertise.

Large Language Models (LLMs) such as Google's LaMDA, HuggingFace's Bloom, and OpenAI's GPT-3 are causing a paradigm shift in Natural Language Processing (NLP), according to a study by Billing *et al.* (2023). Beyond traditional text-based applications, these pre-trained models find use in code generation, copywriting, and product requirement documentation. Notwithstanding the focus on human-robot interaction (HRI), as demonstrated by the humanoid robot Ameca, publicly available LLM implementations for HRI are conspicuously lacking. This result points to a deficiency in the application of language models in human-robot interaction, an issue that should be taken into account in the context of language model applications as a whole.

Four essential software components make up the dialogue system from a technical point of view. The conversation history is maintained by the Chatbot service and is sent to GPT for text completion. The Chatbot bridge bridges the gap in Python versions by allowing the Python 3.10 Chatbot service and the Python 2.7 NaoQi extension modules to communicate with each other through ZeroMQ. The speech recognition module records audio from robots, divides them into segments according to volume thresholds, and then uploads the recordings to Google Cloud for speech-to-text processing. The Dialogue module takes GPT-generated completions, translates them into robot speech, controls the turn-based exchanges, and incorporates preset phrases for particular actions performed by the robot. Although it is intended to be extensible, the current implementation is still in its early stages.

2.6 Ontology-based Knowledge Representation in Robotics

Ontology-based knowledge representation plays a crucial role in enhancing the capabilities of autonomous robots, particularly in the context of goal-oriented tasks and human-robot interaction. Robots that use knowledge representation develop cognitive abilities that allow them to interact and make decisions on their own in a variety of settings. The need for ontologies to represent knowledge about objects and the environment has emerged as a major area of interest in the field of autonomous social robotic systems. Ontologies link individual instances and define their roles in the domain, making it easier to formally specify shared concepts. This method improves adaptability, reusability, and flexibility for a range of robotic tasks, such as planning, manipulation, navigation, and recognition, in diverse environments like offices, homes, and public areas. Through web-based services, knowledge-enabled approaches based on ontology offer a way for different robots to share knowledge and understand semantics, overcoming programming challenges for a variety of tasks and environments. RoboEarth, KnowRob, openEASE, RoboBrain, and RoboCSE are a few notable examples.

Numerous knowledge representation systems have been created to improve autonomous robot performance across a range of applications. As part of the RoboEarth (Waibel *et al.*, 2011)project, which aims to create a World Wide Web of robotics, KnowRob (Beetz *et al.*, 2018) is a reusable framework designed for household manipulation tasks in assistive kitchens. OROSU (Gonçalves & Torres, 2015) is a project that maps robotic ontologies in the medical domain, with a focus on surgery. CARESSES (Bruno *et al.*, 2019) emphasizes cultural competency and interpersonal skills while focusing on assistive robots for senior care. PMK (Diab *et al.*, 2019) improves robotic performance in motion planning and manipulation tasks across a range of scenarios. SARbot (Sun *et al.*, 2019) is a disaster search and rescue robot that uses ontology-based knowledge representation. IEQ (Ribino *et al.*, 2021) uses social humanoid robots to monitor the quality of the indoor environment. Through semantic IoRT systems, SmartRules for the Internet of Robotic Things provides assistance to elderly people in ambient assisted living applications. For human-robot interactive services, ARBI (Chang *et al.*, 2020) uses a knowledge model, and the worker-cobot framework facilitates communication between industrial robots and human workers. APRS brings agility to industrial robots in the kitting domain. An extension of KnowRob, KnowRob 2.0 combines techniques for autonomous robotic agents to render and reason using physics simulation. OROSU integrates healthcare ontologies to focus on robotic assistance in human surgeries. CARESSES provides autonomous robots with cultural competency and communication abilities. By focusing on standardized ontological concepts and reasoning knowledge, PMK contributes to the formulation and reasoning of knowledge. SARbot uses ontology-based decision-making and high-level control to accomplish difficult tasks in disaster situations. IEQ keeps an eye on the comfort and well-being of indoor environments. For context-aware IoRT systems, SmartRules offers context awareness. ARBI presents an ontology-based integrated knowledge model for human-robot interactive services. In cooperative work cells, agile manufacturing is achieved through the worker-cobot framework. APRS uses ontology-based information representation models to give manufacturing robots agility during the kitting process. The state of the art in knowledge representation for autonomous robots across a variety of applications is greatly advanced by these systems.

2.7 Evaluation of Results

The last section of this chapter covers the most common methodologies used to evaluate the proposed interventions in this field. In the study done by Kanda *et al.* (2012), the course consisted of eight two-hour courses that educated children about robotics using Lego Mindstorms.The first five lessons covered basic concepts, such as using one motor and moving on to combining motors, touch sensors, and programming to create an exploration robot. Each lesson was organized into phases, with Robovie, a robot, guiding the students through instructional videos and hands-on tasks. The class included basic tasks to ensure understanding, supplemented by videos and discussions. The advanced tasks required applying learned concepts, like building a robot that can run fast. The six and seven sessions' within-group activities involved cooperative planning, role-playing, and competitive scoring. The final lesson evaluated individual learning as children constructed robots on their own.

Robovie's perception abilities were similar in both conditions, responding to specific spoken keywords; in the social condition, Robovie initiated conversations and praised children's achievements, while in the non-social condition, it judged their performance. The study used a between-participants design, focusing on the controlled factor of social behavior, with two conditions: a non-social condition where Robovie exhibited minimal behaviors for class management, and a social condition where additional social behaviors were displayed. The study was conducted in a research laboratory over eight lessons, each lasting two hours. An adult assistant was available for technical issues, but minimal support was provided. Pretests and posttests, along with short interviews, were conducted to assess learning outcomes, with shorter learning time. For the within-participants factor, a pre-post comparison was used to evaluate learning effects. Robovie's perception capabilities were similar in both conditions, responding to specific spoken keywords. In the social condition, Robovie initiated conversations and praised children's accomplishments, while in the non-social condition, it judged their performance.

The following measurements were taken in the experiment:

1. **Individual Ability(Pretest)**

    - Assessment of math and Japanese skills using a subset of items from examinations that are made available to the public (scores 0–20).

- The time required to build a small Lego car is used to estimate the amount of time spent creating Legos.

2. **Perception of Robot(Posttest)**
   - Strength of Relationship: Robovie's perceived familiarization, likeability, and desire to be liked.
   - Enjoyment: A Likert scale-based assessment of Robovie's level of enjoyment.
   - Impressions: Single-item scales assessing if Robovie is perceived as teacher-like or friend-like.

3. **Social Acceptance(Posttest)**
   - The use of a Heerink et al. scale to assess participants' willingness to use Robovie again, both in a Lego class and in other classes.

4. **Learning Outcome**
   - Basic Knowledge (Pretest and Posttest): Children were asked to create instructions for building an exploratory robot, with the responses coded using a checklist.
   - Basic Achievement (Posttest): Using a ten-item checklist, analyze the Lego robots made during the eighth lesson.
   - Overall Achievement (Posttest): Evaluation of the performance of exploration robots capturing flags, based on the ratio of captured flags.

Charisi *et al.* (2016) conducted a comprehensive review on the evaluation of child-robot interaction in learning settings, focusing on user-centered perspectives and thematic organization based on research objectives. They proposed a conceptual framework, aligning themes such as social interaction, social acceptance, emotional interactions, learning processes, and learning outcomes with corresponding measures. This review highlights current trends, identifies needs specific to the field, and offers valuable insights into methodological directions for evaluating child-robot interaction in educational contexts. Notably, the authors took age ranges into account to ensure the selection of developmentally appropriate evaluation methods. In addition, the authors discuss the limitations and challenges of current methodologies while offering potential future directions for research in this domain.

The conceptual framework created by the systematic coding of 135 eligible studies divides learning objectives into six interconnected thematic constructs:

1. **Social Interaction** This construct especially focuses on the research aim of social characteristics within Child-Robot Interaction (CRI) and includes both short-term and long-term interactions between children and robots, involving both spoken and nonverbal communication.

2. **Social Acceptance** focuses on children's perceptions and attitudes toward the social nature of the robot, indicating their readiness to acknowledge it as a social agent. Social acceptance builds on social interaction by assigning social qualities to the robot, which may result in novel behaviors not observed in other interactions.

3. **Emotional Interaction** This construct studies required emotional statements necessary for emotional engagement, such as the building of trust, in long-term interactions capable of triggering children's emotional engagement and ultimate bonding with the robot.

4. **Learning Process** Represents the learning trajectory shaped by a learner during contact with a robot in the context of a specific learning activity, either in a single session or across time.

5. **Learning Outcome** This concept is defined as the achievement of a particular learning objective, such as the development of desired abilities and competencies or cognitive accomplishments. It is consistent with developmental psychology's identification of developmental stages that represent children's cognitive competence according to age or environmental factors.

6. **Age Appropriateness** When selecting an assessment method, take the age of the children into consideration. A developmentally appropriate assessment method can also be selected based on the child's cognitive ability.

2.7.1 Questionnaires

2.7.1.1 Godspeed Questionnaire Series

The Godspeed Questionnaire Series (GQS) is one of the most highly cited and used questionnaire in the field of Human-Robot Interaction and Human-Agent Interaction (Bartneck *et al.*, 2023). It was developed to measure users' perceptions of robots. The GQS has been translated into 19 languages and is used to assess anthropomorphism (how human-like the robot seems), animacy (how alive the robot appears), likeability, perceived intelligence, and perceived safety.

Anthropomorphism		
Fake	1 2 3 4 5	Natural
Machinelike	1 2 3 4 5	Humanlike
Unconscious	1 2 3 4 5	Conscious
Artificial	1 2 3 4 5	Lifelike
Moving rigidly	1 2 3 4 5	Moving elegantly
Animacy		
Dead	1 2 3 4 5	Alive
Stagnant	1 2 3 4 5	Lively
Mechanical	1 2 3 4 5	Organic
Artificial	1 2 3 4 5	Lifelike
Inert	1 2 3 4 5	Interactive
Apathetic	1 2 3 4 5	Responsive
Likeability		
Dislike	1 2 3 4 5	Like
Unfriendly	1 2 3 4 5	Friendly
Unkind	1 2 3 4 5	Kind
Unpleasant	1 2 3 4 5	Pleasant
Awful	1 2 3 4 5	Nice
Perceived Intelligence		
Incompetent	1 2 3 4 5	Competent
Ignorant	1 2 3 4 5	Knowledgeable
Irresponsible	1 2 3 4 5	Responsible
Unintelligent	1 2 3 4 5	Intelligent
Foolish	1 2 3 4 5	Sensible
Perceived Safety		
Anxious	1 2 3 4 5	Relaxed
Calm	1 2 3 4 5	Agitated
Still	1 2 3 4 5	Surprised

Table 2.1: Godspeed Questionnaire Series

2.7.1.2 Smileyometer

The Smileyometer is a popular subjective measure for measuring enjoyment, especially among children. (van der Sluis *et al.*, 2012) It's essentially a smiley-faced variation of a 5-point Likert scale that measures various aspects of fun (Figure 2.1). Smiley Face Likert scales are used as a rating scale for quantitative questions in evaluations by the Smileyometer. When working with children, this strategy is very beneficial because they are typically more capable and motivated by using faces rather than words or numbers.

The Smileyometer has been utilized in a variety of scenarios, including assessing children's fun and enjoyment in a museum and measuring product liking in preschool children. In essence, the Smileyometer allows children to convey their feelings about an experience in a visually attractive and intuitive way, making it a significant tool for researchers in domains such as education, psychology, and human-computer interaction.

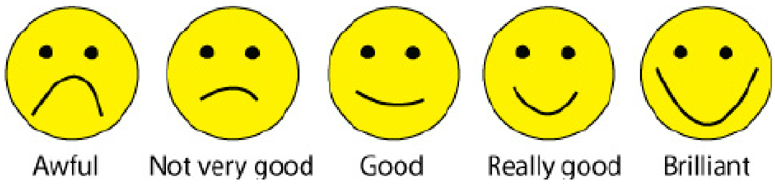

Figure 2.1: Smileyometer Rating Scale

Chapter 3

Methodology

3.1 System Architecture Overview

The central focus of this thesis is the exploration of intercultural pedagogy through the use of a social robot, specifically the Nao robot.The Nao robot is programmed to narrate stories to children in their native language, creating a familiar and engaging atmosphere.

The major goal of these narratives is to familiarize children with different cultures in a social environment. The stories are centered on the robot introducing itself as an extraterrestrial being, sparking the children's interest and involvement. Throughout the story, the robot asks the children questions like, "I come from this alien planet, where are you from?" The children's responses, which may range from "I am from Lagos, Nigeria" to "Io sono Italiano" or "Ich komme aus Berlin," give an environment for cultural interchange and learning.

The system uses Google's Speech-to-Text API, which can recognize many languages, to process the children's responses. The responses of the children are then evaluated by the OpenAI API to generate appropriate responses or follow-up questions from the robot.

In addition to these tools, the system includes a knowledge base ontology written in Protégé. This ontology contains information about numerous countries and their customs, greetings, music, food, and more. Using this knowledge base, the robot can generate comments or questions that are relevant to the child's prior response, keeping the discussion flowing and improving the learning experience.

The system also makes use of Google Cloud APIs to allow the robot to understand and reply to inquiries and replies in the children's native languages. This feature considerably improves the system's accessibility and effectiveness, making it an effective tool for developing cultural awareness and inclusivity among children.

In summary, this thesis describes a comprehensive system that combines the characteristics of a social robot with advanced language processing tools and a rich knowledge base to facilitate intercultural learning among children. The system shows how technology may be used to improve teaching methods and promote a more inclusive and culturally conscious society.

All the code related to the thesis project can be accessed at this link: https://github.com/jerin-joy/social_robot_for_intercultural_pedagogy

3.2 System Components

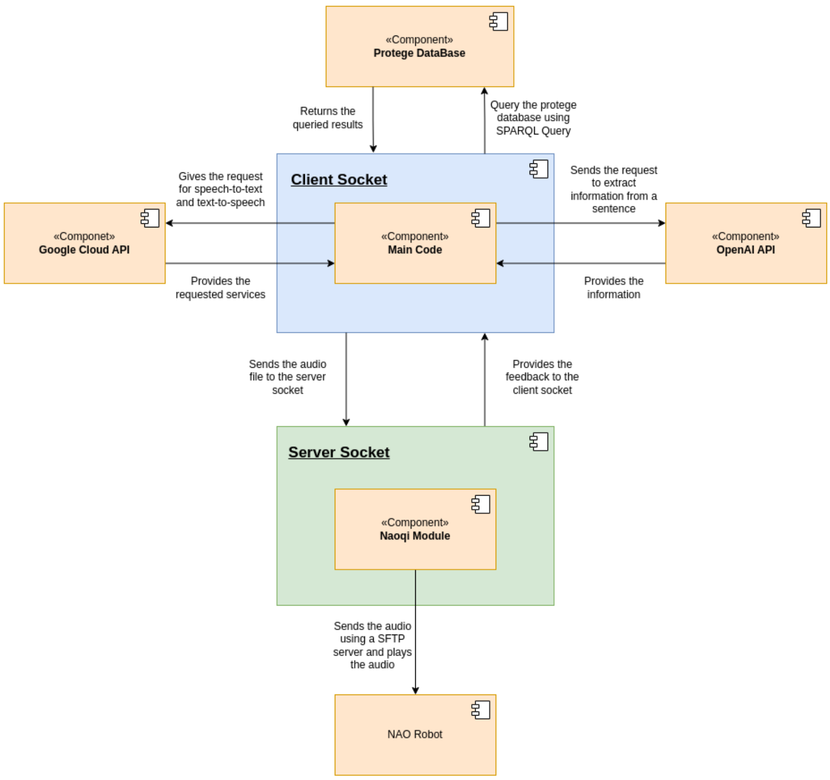

Figure 3.1: Component diagram

The primary system architecture is composed of three components: a client socket, a server socket, and the Nao robot. The client socket houses the main code, which controls the other components, including the Google Cloud API, OpenAI API, and Protégé Database.

The main code sends requests for services such as speech-to-text, text-to-speech, and cloud translation to the Google Cloud API, which in turn provides the requested services. Similarly, the main code sends a request to the OpenAI API to extract information from a sentence, and the OpenAI API provides the required information. The main code also queries the Protégé Database using a SPARQL query, and the database returns the queried results.

In addition, the client socket sends the translated text and the language code to the server socket via a socket connection. The server socket accesses the audio file processed by the Google Cloud text-to-speech API and sends it to the Nao robot using an SFTP server. The Nao robot then plays the audio. Once the audio has been played by the Nao robot, the server socket sends feedback to the client socket via the socket connection.

The architecture of the project is primarily maintained by two versions of Python: Python 3 and Python 2. The main codebase, which comprises the core functions and logic, is written in Python 3. This Python version is compatible with the most recent libraries and frameworks, making advanced features easy to implement.

Python 2 is primarily used for the NAOqi module, which is built inside the Nao robot. NAOqi is the primary software that operates the robot and serves as its programming framework. This framework facilitates communication between multiple modules (motion, audio, and video) and supports parallelism, resources, synchronization, and events.

To ease communication between the Python 3 main codebase and the Python 2-based NAOqi module, a socket connection is employed. This connection serves as a bridge, allowing the two Python environments to communicate and cooperate. The socket connection is critical for the overall system's seamless operation since it enables real-time communication and data transfer between the controlling system and the robotic platform.

3.2.1 Nao Robot

The Softbank Nao robot is a small autonomous, and programmable humanoid robot created by Aldebaran Robotics, a French robotics company acquired by SoftBank Group and rebranded as SoftBank Robotics in 2015.

Nao is equipped with a multitude of sensors and can walk, dance, speak, and recognize faces and objects. It features 25 degrees of freedom and the ability to walk. For better understanding the environment, it also features two 2D cameras, one bumper on each foot, four microphones, and seven touch sensors.

Nao is utilized in research, education, and healthcare all across the world. It is used by businesses and healthcare facilities to welcome, inform, and entertain guests, as well as as a programming tool in education and research. Nao assists teachers in bringing teachings to life, making learning more enjoyable and concrete for kids. It's also utilized in research to run interactive experiments, collect data, and assist researchers in testing new theories and finding new answers.

Figure 3.2: Nao Robot

Nao is fluent in over 20 languages, allowing it to speak effectively and easily with students from all cultures. It is built to move naturally, identify obstacles, prevent falls, and get right back up.

Nao's development began in 2004 with the launch of Project Nao. Since then, various versions of the robot have been developed, including the Nao Academics Edition for universities and laboratories, as well as the Nao Next Gen and Nao Evolution, both of which include hardware and software upgrades.

In conclusion, the Softbank Nao robot is a versatile and interactive tool that is extensively used in a variety of disciplines due to its capabilities in movement, speech, and interaction, making it a significant asset in education, research, and healthcare.

3.2.2 Software & Tools

The main software and tools used in this thesis project are Protégé, Google Cloud API, and OpenAI API.

3.2.2.1 Protégé

Protégé is a free and open-source ontology editor and knowledge management system. It was created in 1987 by Mark Musen and has since been improved by a team at Stanford University. Protégé is the most popular and commonly used ontology editor in the world. It provides a set of tools for building domain models and knowledge-based applications using ontologies. Protégé is backed by a large network of academic, government, and business users that utilize it to create knowledge-based solutions in fields as diverse as biomedicine, e-commerce, and organizational modeling.

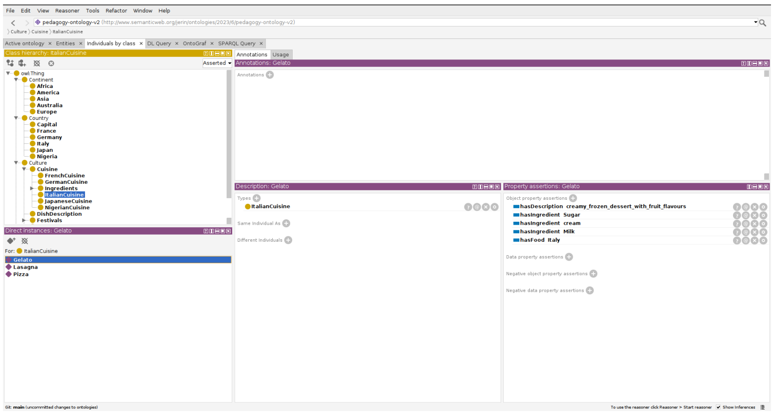

Figure 3.3: Protégé Software

3.2.2.2 Google Cloud API

Google Cloud Speech-to-Text API: This is a Google cloud-based tool that allows developers to translate spoken language into written text. It supports several languages and audio formats and has highly accurate voice recognition capabilities. Using strong neural network models in an easy-to-use API, developers may convert audio to text in over 125 languages and variants.

Google Cloud Text-to-voice API: This allows developers to generate human-like speech from text. The API translates text to audio in a variety of formats and supports a variety of languages, voices, and speech parameters. Speech Synthesis Markup Language (SSML) can also be used by developers to tailor the speech output.

Google Cloud Translation API: This API allows your websites and applications to dynamically and programmatically translate text. To translate text, it employs either a Google pre-trained or a customized machine learning model.

The API can translate text between thousands of language pairs in real time.

3.2.2.3 OpenAI API

The OpenAI API is a cloud interface hosted on Microsoft Azure. Users can obtain newly created pre-trained AI models from OpenAI, including DALL-E, Codex, and GPT-3. Modern AI capabilities can be added to almost any work thanks to the OpenAI API's optimal architecture. In contrast to conventional AI systems, which are often created for a single use case, the OpenAI API offers developers a versatile cloud platform for text-in and text-out applications.

3.2.3 Software Dependencies

The implementation of this project required the use of several Python packages. The following is a list of the main packages and their versions used in this project:

Package	**Description**
google-cloud	Used for the Google Cloud services, including the Text-to-Speech and Speech-to-Text functionalities.
openai	Used to interact with the OpenAI API for information extraction.
owlready2	Used to load the ontology and run SPARQL queries.
pydub	Used for handling audio files.
socket	Used to establish a socket connection between the main code and the Nao robot.
speech_recognition	Used for recognizing speech.
google-cloud-translate-v2	Used for the Google Cloud Translation API.
google-cloud-speech-v2	Used for the Google Cloud Speech-to-Text API.
google-cloud-texttospeech-v1	Used for the Google Cloud Text-to-Speech API.

Table 3.1: List of Python packages and their descriptions

3.3 System Workflow

The temporal diagram (3.4) below shows the overall workflow of the project.

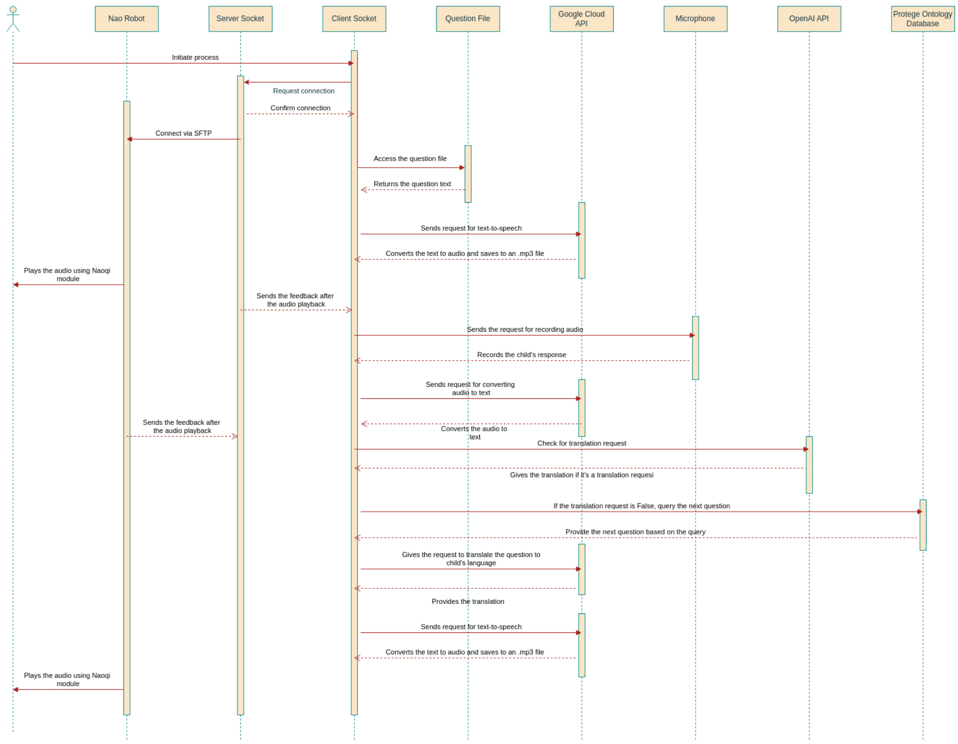

Figure 3.4: Temporal diagram

The main code sends requests for services such as speech-to-text, text-to-speech, and cloud translation to the Google Cloud API, which in turn provides the requested services. Similarly, the main code sends a request to the OpenAI API to extract information from a sentence, and the OpenAI API provides the required information. The main code also queries the Protégé Database using a SPARQL query, and the database returns the queried results.

In addition, the client socket sends the translated text and the language code to the server socket via a socket connection. The server socket accesses the audio file processed by the Google Cloud text-to-speech API and sends it to the Nao robot using an SFTP server. The Nao robot then plays the audio. Once the audio has been played by the Nao robot, the server socket sends feedback to the client socket via the socket connection.

Here is a breakdown of the processes into several sections:

3.3.1 Setting Up The Connection

1. As the first step, the connection between the client socket and the server socket is initiated by executing both codes in separate terminals.

3.3.1.1 Client-Server Communication

The client-server communication paradigm is a key component of the system design, which enables smooth communication between various components. The client, which runs on Python 3, contains the main codebase responsible for processing and creating questions, as well as handling user interactions. The Python 2 server is responsible for overseeing the communication with the Nao robot and carrying out commands on its behalf.

A socket connection is used for communication between the client and server. The server can receive requests and instructions from the client through this bidirectional connection, and the client can receive information back from the server. In order to provide real-time data interchange and enable coordination between the robotic platform (server) and control system (client), a socket connection is essential.

Both Python 2 and Python 3 are used in the project to take advantage of their respective advantages for particular functionalities. The primary codebase, written in Python 3, makes use of the libraries, frameworks, and modern features available in the language. Python 3 guarantees interoperability with the most recent advancements in software engineering and artificial intelligence.

However, Python 2 is used to implement the Naoqi module, which communicates with the Nao robot. The Naoqi framework's compatibility requirements, which are more in line with Python 2, force this decision. The socket connection that has been established allows for easy management of the dual-language approach and facilitates efficient communication between the main Python 3 codebase and the Python 2-based Naoqi module.

2. Using the username and password of Nao, the server socket connects to the Nao robot over an SFTP connection.

3.3.1.2 SFTP Connection

The server and the Nao robot connect securely via the use of Secure File Transfer Protocol (SFTP). Transferring questions, audio files, and other data between the robotic platform and the controlling system depends on this connection. Asserting the confidentiality and integrity of the data being transferred, SFTP is compliant with industry standards for secure communication in distributed systems.

In conclusion, the system architecture ensures secure data transfer via SFTP connections and is built to smoothly integrate the main codebase, the Nao robot, and the Python 2-based Naoqi module through client-server communication. The two languages work together harmoniously when using Python 2 and Python 3, which makes it possible to complete tasks on the robotic platform and control system with efficiency.

3.3.2 Question Preparation

1. Questions are saved in a separate Python file and are assigned in order.

3.3.2.1 Question Database

The question database consists of various questions based on Intercultural pedagogy. The questions database acts as a repository for a collection of diverse and engaging questions that are thoughtfully designed to provoke answers from the user, encouraging a creative and participatory experience. The database is organized as a list, with each question intended to spark participants' interest and encourage thoughtful replies from the participants. The questions themselves have an imaginative and exploratory tone, which is influenced by the narrator Robo's fantasy world on the planet Zogar.

The questions are intended to gradually reveal a narrative while inviting users to share their perspectives and experiences. Themes include learning about other cultures, traveling to other planets, and discovering natural wonders. Users are asked to consider everything from ways of communicating to foods they like, spending time with animals, and recreational activities in their home countries.

2. To enhance engagement, the names of the children participating in the experiment are assigned randomly in the questions to introduce a touch of personalization. The questions database's structure is designed to be flexible and adaptable, allowing for a wide range of responses and guaranteeing a lively and insightful dialogue with users. In keeping with the project's main objective of developing an interactive and culturally inclusive platform, this careful selection of questions seeks to offer a distinctive and enjoyable user experience.

3.3.3 Text-to-Speech Conversion and Audio Playback

1. The first question from the list of questions is picked and asked in English by default.

2. The question text is converted to audio using Google Cloud Text-to-Speech API.

3.3.3.1 Google Cloud Text-to-Speech API

The system's text-to-speech conversion is managed by a specific function called `synthesize_speech`, which makes use of the Google Text-to-Speech (TTS) API. The following section explains the main elements of this code and how it converts a subset of the question text into audio:

(a) **Speech Synthesis Initialization:**
The method starts by initializing a `TextToSpeechClient`, which is an instance of the Google Text-to-Speech API client. The purpose of this client is to simulate speech by interacting with the API.

```
speech_client = texttospeech_v1.TextToSpeechClient()
```

(b) **Text Translation:**
Before speech synthesis, a separate translation function `translate_text` is used to translate the original question text into the target language. This makes sure that the synthesized speech is coherent with the intended linguistic context.

```
translated_text = self.translate_text(target_language,text)
```

3. The voice is assigned and it is saved to an MP3.

(a) **Speech Synthesis Configuration:**
The `synthesize_speech` function sets up the parameters for voice selection and audio encoding, as well as the synthesis input, by designating the translated text as the input for speech generation.

(b) **Speech Synthesis Request:**
A request is sent to the Google Text-to-Speech API `speech_client.synthesize_speech` using the provided parameters, which include the synthesis input, voice selection, and audio configuration.

Algorithm 1 Text-to-Speech Conversion

1: Initialize the TextToSpeechClient as *speech_client*
2: Create a SynthesisInput object with the translated text
3: **if** the target language is 'en-US' **then**
4: Create a VoiceSelectionParams object with the language code as the target language, the SSML gender as female, and the name as 'en-US-Wavenet-F'
5: **else**
6: Create a VoiceSelectionParams object with the language code as the target language and the SSML gender as female
7: **end if**
8: Create an AudioConfig object with the audio encoding set to MP3

```
response1 = speech_client.synthesize_speech(
    input=synthesis_input,
    voice=voice1,
    audio_config=audio_config
)
```

(c) **Audio File Generation:**
The API response's synthesized audio material is then written to an MP3 file called "audio.mp3."

```
with open('audio.mp3', 'wb') as output:
    output.write(response1.audio_content)
```

4. The MP3 is then converted to WAV in order to make it compatible with Nao.

3.3.3.2 Audio Format Conversion

The resulting MP3 file is converted to WAV format using the pydub package to ensure compatibility with the Nao robot's audioplayer.

```
sound = AudioSegment.from_mp3("audio.mp3")
sound.export("audio.wav", format="wav")
```

The entire text-to-speech conversion process is encapsulated in this comprehensive code structure, which includes translation, voice selection, and audio file generation. The resulting "audio.wav" file is then ready for playback on the Nao robot, enriching the user experience with synthesized speech tailored to the specified language and voice parameters.

5. The question text and the language code are sent to the server socket to print in the terminal as a receipt.

6. The server socket receives the text and prints it to the terminal.

7. The received text messages are saved in a log file with a date and time stamp.
   Logging: For the purpose of capturing and timing interactions between the robot and the child, the `ConversationLogger` class in the `conversation_log.py` file offers an organized method for logging messages in the conversation. The format of the logs contains the timestamp, sender (either 'Robot' or 'Child'), and the message exchanged. The logging functionality is explained below.

Algorithm 2 Logging

Import the datetime module
Define class ConversationLogger
Define `__init__` **method** with parameters `self` and `filename`
Inside the `__init__` **method**:
    Assign `filename` to `self.filename`
Define `log_message` **method** with parameters `self`, `sender`, and `message`
Inside the `log_message` **method**:
    Open the file with name `self.filename` in append mode as `file`
    Get the current date and time, format it as "YYYY-MM-DD HH:MM:SS", and assign it to `timestamp`
    Write a string to the file in the format "timestamp - sender: message"

Usage

For logging messages from the robot section:

```
logger = ConversationLogger('conversation.log')
logger.log_message('Robot', sentence)
```

For logging messages from the child section:

```
logger = ConversationLogger('conversation.log')
logger.log_message('Child', sentence)
```

This logging system helps to preserve a conversation log in chronological order, makes analysis easier, and sheds light on the dynamics of the robot-child relationship.

8. The WAV file is then copied inside the Nao robot using an SFTP server.

9. Nao uses the audioplayer in the Naoqi package to play the audio file.

10. It delivers feedback that the audio has completed playing after it has done playing.

11. The feedback is sent to the client socket by the server socket.

3.3.4 Recording and Transcribing the Child's Response

1. Upon receiving the sentence, the client socket records the child's response through the microphone, which ceases recording once the child stops speaking. The audio is saved as an MP3 file and then converted to text using the Google Speech-to-text API, which also detects the spoken language from the audio. The system then returns the converted text along with the language code.

   Speech Recognition: The two primary components of the system's voice recognition functionality are audio recording and speech-to-text conversion via the Google Cloud speech-to-text API. A detailed explanation of each component is provided below:

   Audio Recording

   Using the SpeechRecognition library, the `record` method records audio input through the system's microphone. The audio recording procedure is as follows:

   Usage

   The usage of the function is as follows:

Algorithm 3 Recording Audio

function RECORD($self, audio\_file$)
  Inside the RECORD **method**:
    Initialize a Recognizer object and assign it to r
    Open the default microphone as *source*
    Adjust the recognizer for ambient noise using *source*
    Print "Please say something..."
    Listen to the *source* and assign the audio to *audio*
    Open the file with name $audio\_file$ in write binary mode as f
    Write the wav data of *audio* to the file
end function

```
translator = SpeechToTextTranslator(
   project_id, language_codes,
   audio_file
)
translator.record(audio_file)
```

Initialization: The `project_id`, `language_codes`, and `audio_file` parameters are used to initialize the `SpeechToTextTranslator` class. The `project_id` is the project identifier that the translator is being used for. The `language_codes` parameter gives the codes of the languages that need to be translated. The `audio_file` is the file that had been recorded by the microphone and needs to be translated.

Recording: The `record` method of the `SpeechToTextTranslator` class is called with the `audio_file` as a parameter. This method adjusts for ambient noise, prompts the user to say something, records the audio, and then saves it to the specified file in WAV format.

3.3.4.1 Speech-to-Text Conversion

Here's a comprehensive overview of how the `transcribe_multiple_languages_v2` method turns recorded audio into text using the Google Cloud Speech-to-Text API:

(a) **Client Instantiation**: A client for the Speech-to-Text API is created using `client = SpeechClient()`.

Algorithm 4 Speech-to-Text Conversion

```
function TRANSCRIBE_MULTIPLE_LANGUAGES_V2(self)
    client ← SPEECHCLIENT
    Open the file with name self.audio_file in read binary mode as f
    content ← Read the content of the file
    config ← RECOGNITIONCONFIG(auto_decoding_config = AutoDetectDecodingConfig, language_codes = self.language_codes, model = "latest_long")
    request ← RECOGNIZEREQUEST(recognizer = "projects/{self.project_id}/locations/global/recognizers/_", config = config, content = content)
    response ← CLIENT.RECOGNIZE(request)
    if response.results is not empty then
        for each result in response.results do
            return Transcript of the first alternative of the last result, Language code of the last result
        end for
    else
        return None, None
    end if
end function
```

(b) **File Reading**: The audio file specified by `self.audio_file` is opened and read as bytes; the content of the file is stored in the `content` variable.

(c) **Configuration Setup**: A `RecognitionConfig` object is created with several parameters:

- `auto_decoding_config`: This is set to `AutoDetectDecodingConfig()`, which automatically detects and decodes the audio.
- `language_codes`: This is set to `self.language_codes`, which should be a list of language codes denoting the languages to be used for transcription.
- `model`: This is set to `"latest_long,"` which determines the transcription model to be used.

(d) **Language Detection** The code that recognizes the user's spoken language is located in the `RecognizeRequest` and processing of the `response`.

```
        request = cloud_speech.RecognizeRequest(
        recognizer=f"projects/{
        self.project_id
        }/locations/global/recognizers/_",
        config=config,
        content=content,
    )
```

The `config` parameter is set to a `RecognitionConfig` object, which includes `language_codes=self.language_codes` indicating that the Speech-to-Text API will attempt to identify speech in the languages indicated by `self.language_codes`.
In the processing of the `response`:

```
if response.results:
for result in response.results:
    pass
transcript = result.alternatives[0].transcript
language_code = result.language_code
return transcript, language_code
```

If the `response` has results, the method iterates over the results, returning the transcript of the last result along with its `language_code`. Each `result` in `response.results` is a `SpeechRecognitionResult` object, which includes a `language_code` attribute that represents the language in which the speech was recognized.

(e) **Result Processing**: If the response contains results, the method iterates over them and provides the transcript and language code of the final result; otherwise, the method returns `None`, `None`.

Usage

The usage of the function is as follows:

```
translator = SpeechToTextTranslator(
    project_id, language_codes, audio_file
)
transcribed_text = (translator.transcribe_multiple_
                    languages_v2()[0])
og_language = (translator.transcribe_multiple_
               languages_v2()[1])
```

2. The converted text is saved to a log file.

3.3.5 Handling Translation Requests and Preparing for the Next Question

1. It checks a list of keywords to see if it's a translation request.
2. If it is a translation request, it uses the OpenAI API to provide the translation of the previous question in the child's language.

3.3.5.1 OpenAI API

A key element of the system is the `InformationExtractor` class, which interfaces with the OpenAI API to extract data in response to user prompts. Contextually relevant responses are generated by the class using the ChatCompletion feature, which makes use of the sophisticated language model GPT-3.5-turbo. The main objective is to apply natural language processing to improve the system's comprehension and response to user inputs.

An `extract_information` method in the class encapsulates the communication with the OpenAI API. This technique enables the dynamic extraction of information by accepting input text and a user prompt. The method includes a retry mechanism with adjustable parameters to improve reliability and guarantee consistent performance even in the event of potential API errors or timeouts.

The main steps for initializing the class and running the `extract_information` method are shown in the pseudocode that follows. These procedures show how the `InformationExtractor` uses a systematic approach to leverage the OpenAI API's capabilities for system-wide information extraction.

Translation Request Handling

After each response by the child, the code does periodic checks for text translation. This is done to guarantee that the translated question is accessible to all children who speak different languages. For example, if a German-speaking child does not understand the question in Italian, he might request that it be translated into German. The code then looks through a list of words in each language that sound similar to the word 'translate' in their respective language, and if the child speaks any of the 'translate' words, it will send to the OpenAI translation request function, which will then translate the sentence.

Explanation

- The dictionary `translation_keywords` contains translation keywords for many languages.
- The `is_translation_request` function determines whether a translation keyword appears in the transcribed text. If it finds one, it calls the `translation_request` function and returns True. If not, it returns False.
- Translation requests are handled via the `translation_request` function. It prints the mapped language name, generates a prompt, uses the `InformationExtractor` to extract information, synthesizes speech, and sends it to Nao.

3. If the translation request is false, the code prepares for the next question.

4. The OpenAI API is used to extract the required information (for example, a country name from the sentence) for the next question from the transcribed text.

Algorithm 5 Information Extraction

Initialize an instance of the `InformationExtractor` class.

function InformationExtractor
    Set up OpenAI API credentials using the provided API key.
end function

function extract_information($text, prompt, temperature = 0.5, max\_retries = 5, timeout = 5$)
    Input: *text* (the input text for information extraction), *prompt* (the user's prompt), *temperature* (optional, default value is 0.5), *max_retries* (optional, default is 5), *timeout* (optional, default is 5 seconds).
    for i **in range** *max_retries* **do**
        Make a ChatCompletion request to the OpenAI API:
            – Model: "gpt-3.5-turbo"
            – Messages: A list with a single message object representing the user's prompt.
            – *max_tokens*: Set to 1024 to handle longer responses.
            – *n*: Set to 1 to get a single response.
            – *stop*: Set to None to allow the model to generate a complete response.
            – *temperature*: Controls the randomness of the response (default is 0.5).
            – *timeout*: Add a timeout for the request (default is 5 seconds).
        Extract the generated content from the response.
        Return the extracted information.

        if Exception (OpenAIError or Timeout) **then**
            Print a message indicating that the request failed or timed out and is being retried.
            Wait for 1 second before retrying.
        end if
    end for

    if the loop completes without successfully obtaining a response then
        Raise an exception.
    end if
end function

Algorithm 6 Translation Request Handling

```
Define translation_keywords dictionary with translation keywords for different languages.
      transcribed_text, og_language, text_to_be_translated
function IS_TRANSLATION_REQUEST(Parameters)
    for each translation keyword for the original language (og_language) do
        if a translation keyword is found in transcribed_text then
            Print "It's a translation request".
            Call translation_request with transcribed_text, text_to_be_translated, and og_language.
            Return True.
        end if
    end for
    If no translation keyword is found:
        Return False.
end function
transcribed_text, text_to_be_translated, language_code
function TRANSLATION_REQUEST(Parameters)
    Set target_language as language_code.
    Map language_code to its corresponding language name (English, Italian, German).
    Print the mapped language name.
    Create a prompt for information extraction:
        "Child asked robot: Can you translate {text_to_be_translated}. Translate ONLY('{text_to_be_translated}') in {language_code}"
    Use the InformationExtractor class to extract information based on the prompt.
    Print the extracted response.
    Use the translator to synthesize speech in the target language.
    Send the synthesized speech to Nao with the appropriate language code.
end function
```

User Interaction and Response handling

The child's responses to specific questions are sent to several functions that use the OpenAI API to extract information from the child's responses. The following pseudocode explains one of the many functions:

Explanation for `get_response_make_dish`

- This function was designed to handle responses relating to preparing dishes at home.
- It extracts information using a predefined prompt.
- The `InformationExtractor` class is used to extract information from user responses.
- The retrieved response is then synthesized into speech and printed.

5. To obtain the elements associated with the received information, the relevant information is queried in the Protégé ontology database using a SPARQL query. (For instance, country capital)

3.3.5.2 Knowledge Representation and Ontology Structure

Protégé software is integrated with the project to enable item queries based on requirements. Intentional question design ensures a logical flow between questions, enabling questions to be contextually related to previous ones. The system uses the OpenAI API to extract relevant information from the child's answers to earlier questions. This data is then used to formulate targeted queries in the Protégé database to retrieve the required data. This method improves the system's ability to participate in meaningful discussion by modifying its queries based on the context offered by the child's responses.

After data extraction via the OpenAI API, the system queries the Protégé ontology database. In this process, the information extracted via the OpenAI API is utilized as a key to retrieve relevant information that is stored in the ontology. For instance, in the event that the child speaks Italian, the spoken language is identified by the Google Cloud API and passed to the SPARQL Query function. It searches the Protégé ontology database for instances related to the information retrieved from the OpenAI prompt. For example, if a child is speaking Italian, the SPARQL query function looks for instances related to Italy or the Italian language.

Algorithm 7 Response Handling

```
function GET_RESPONSE_MAKE_DISH(transcribed_text)
    Create a prompt for information extraction:
        "The child was asked: 'Have you ever helped make _____(a dish) at home?'. The child replied: '{transcribed_text}'. Give a reply to the child's answer without asking a question."
    Use the InformationExtractor class to extract information based on the prompt.
    Print the extracted response.
    Synthesize speech in the original language using the translated response.
    Print the synthesized response.
end function

function OTHER FUNCTIONS BRIEFLY DESCRIBED
    get_response_communication: Handles responses related to communication.
    get_response_food: Handles responses related to food preferences.
    get_response_try_dish: Handles responses related to trying different dishes.
    get_response_capital: Handles responses related to knowledge of country capitals.
    get_response_animals: Handles responses related to animals and their uses.
    get_response_sport: Returns only the name of a sport mentioned by the child.
    get_response_sport2: Handles responses related to sports and players.
    get_response_adventure: Handles responses related to adventurous wishes.
    get_response_festivals: Handles responses related to participation in festivals.
    get_response_fun: Handles responses related to activities for fun, with special handling for sports-related answers.
    get_response_yes_or_no: Handles responses that expect a 'Yes' or 'No' answer.
    get_country: Extracts the country name mentioned in the transcribed text.
end function
```

The approach guarantees a quick and easy way to retrieve knowledge. The questions' well-structured design and the OpenAI API's dynamic adaptation help to create a conversational flow that makes sense and is relevant to the context. The way information extraction, database querying, and question design work together harmoniously is indicative of the project's methodical and deliberate approach.

Protégé Database

The Protégé database used in this study has a hierarchy of subclasses and object characteristics that all come under the primary class of owl:Thing. This hierarchical structure enables a thorough and organized depiction of the data. For more details about the structure of the Protégé database, refer to the Appendix A

Class Hierarchy

The database covers a wide range of topics, including geographical regions, cultural features, languages, and time of day. The database is flexible enough to include data from various continents and nations, even though the main concentration is on Europe, particularly Italy.

The main topics covered in the ontology can be divided into two components: Geographical Hierarchy and Cultural Hierarchy.

Geographical hierarchy: It contains classes about different continents and countries.
Cultural hierarchy: It contains various classes related to cultures such as Cuisine, Festivals, Greetings, Sport, and various dish descriptions. Cuisine class includes various country-specific cuisines and their ingredients. Likewise, Greetings, Festivals, and Sport classes contain different subclasses relating to the country.

Object Properties

This study's ontology makes use of several object properties, each with its domains and ranges. These properties help the ontology's relationships between different classes and subclasses. The ontology includes elements such as continents, countries, cuisines, dish descriptions, players, greetings, ingredients, languages, time of day, translations, festivals, and sports. It also includes specific elements for German and Italian translations. The object properties serve as the ontology's foundation, allowing meaningful relationships to be established between distinct classes and subclasses.

Direct Instances and Object Property Assertions

Each subclass has numerous direct instances, and each of these instances encompasses multiple property assertions. The Country subclass includes instances representing various countries, including France, Germany, Italy, Japan, and Nigeria. Each country instance is associated with relevant information through object property assertions. Instances under Cuisine subclass represent French, German, and Italian Cuisines, each with specific dishes and ingredients. Instances under DishDescription represent descriptions of various dishes. Instances within the GermanFestivals and ItalianFestivals subclasses represent festivals celebrated in Germany and Italy, respectively. Greetings subclass includes various greetings in French, German, and Italian, each associated with specific times of the day. Instances under GermanSport and ItalianSport represent sports popular in Germany and Italy, respectively. Instances within the Player subclass represent individual players associated with specific sports and countries. Each player instance is intricately linked to its nationality and sport through object property assertions. The TimeOfDay subclass includes instances representing different times of the day.

3.3.5.3 SPARQL Query

The `sparql_query_questions_class.py module` plays a key role in generating dynamic and contextually relevant questions for system interaction. This class is intended to query the Protégé ontology database using SPARQL queries in order to extract information that can be used to generate interesting and culturally appropriate questions for users.

Initialization

With the Owlready2 library, the class constructor loads the ontology from the given directory to initialize it. This ontology acts as a knowledge base with details on greetings, foods, festivals, sports, athletes, and other cultural characteristics.

```
def __init__(self, ontology_path):
self.ontology = get_ontology(ontology_path).load()
```

Query Execution

Based on the given country and time, the `run_query` method runs a SPARQL query to obtain information about greetings and food. After that, the results obtained are analyzed to create dynamic questions with cultural components.

Algorithm 8 Run Query

function RUN_QUERY(*self*, *country*, *time*)
    Construct a SPARQL query string with the given *country* and *time*.
        Use PREFIX to define the ontology namespace.
        Use SELECT to retrieve variables *greeting* and *food*.
        Use WHERE to specify the conditions for the query:
        - *greeting* must haveCountry equal to the provided *country*.
        - *greeting* must haveTimeOfDay equal to the provided *time*.
        - *food* must haveFood equal to the provided *country*.
        Format the query string with the provided *country* and *time*.
    Execute the SPARQL query:
        Use `default_world.sparql` method with the constructed query.
        Store the query results in the variable *results*.
    Call the method `process_greeting_and_food` with the query results and the provided *country*.
        Pass *results* and *country* as parameters to the `process_greeting_and_food` method.
    Return the result of `process_greeting_and_food`.
end function

6. A question is generated using the retrieved elements from the ontology.

Question Generation

After extracting pertinent data from the query results, the `process_greeting_and_food` method creates questions dynamically using pre-established question formats. With placeholders for a greeting, random dish, ingredients, and nation included in these formats, the questions become more varied and culturally rich.

Algorithm 9 Process Greeting and Food

function PROCESS_GREETING_AND_FOOD($self, results, country$)
    Convert *results* to a list and store it in *results_list*.
    Get the greeting from the first result:
        Access the `name` attribute of the first element in the first column of *results_list*.
        Replace underscores with spaces and store the result in the variable *greeting*.
    Get all food items from the results:
        Create a list of food names by extracting the `name` attribute from the second column of each row in *results_list*.
        Store the list of food names in the variable *foods*.
    Select a random food item:
        Use the `random.choice` function to randomly select a food item from the *foods* list.
        Store the selected food item in the variable *random_food*.
    Get the ingredients for the random food item:
        Call the method `get_ingredients` with the *random_food* as a parameter.
        Store the result in the variable *ingredients*.
    Define a list of question formats:
        Include different question formats as strings in the list.
    Select a random question format:
        Use the `random.choice` function to randomly select a question format from the *question_formats* list.
        Store the selected question format in the variable *question_format*.
    Generate the text:
        Format the question using the selected question format, greeting, *random_food*, *ingredients*, and *country*.
        Return the formatted question and the *random_food*.
end function

This pseudocode outlines the steps performed in the `process_greeting_and_food` method, from extracting information from the SPARQL query results to generating a formatted question.

Additional Functionality

Additionally, the class offers ways to find out information about ingredients, the capital of a country, player details, festival details, and more. All these techniques work in a similar way: they run SPARQL queries, handle the output, and produce thought-provoking and interesting questions or statements.

The entire code can be accessed here: https://github.com/jerin-joy/social_robot_for_intercultural_pedagogy/blob/main/questions.py

- The `get_country_capital` method accepts a country and a time as inputs and returns the capital of the country along with a greeting associated with the input country and time via a SPARQL query. The `process_country_capital` method is then used to process the results.
- The `process_country_capital` method generates a statement about the capital of the given country based on the findings of the `get_country_capital` method.
- The `generate_question` method creates a question regarding the traditional preparation of a food item in a given country based on an input of a food item, its ingredients, and the country.
- The `get_description` method accepts a food item as input and returns the food item's description by executing a SPARQL query.
- The `get_phrases` method accepts a language as input and returns sentences and their translations in the specified language via a SPARQL query.
- The `get_main_players` method retrieves the main players in a given sport in a specific country by executing a SPARQL query with two inputs: a sport and a country. The `process_main_players` method is then used to process the results.
- The `process_main_players` function generates a statement about the key players of the given sport in the specified country using the outcomes of the `get_main_players` method.
- Using a country as input, the `get_main_festivals` method conducts a SPARQL query to retrieve the country's main festivals. The `process_main_festivals` function is then used to process the results.

- The `process_main_festivals` method creates a question regarding the major festivals in the specified nation using the outcomes of the `get_main_festivals` method.

With the help of these methods, one can engage with the ontology in an organized manner and obtain particular data according to the input criteria. It is possible to retrieve data from the ontology in a flexible and effective manner by using SPARQL queries.

7. Using Google Translate API, the inquiry is translated to the child's spoken language (the language he spoke previously).

Translation

The code provided includes a `translate_text` method that uses the Google Cloud Translation API to translate text between languages. Here's an overview of the language translation process.

Algorithm 10 Text Translation

```
function TRANSLATE_TEXT(self, target_language, ontology_text)
    Inside the TRANSLATE_TEXT method:
        Initialize a Client object from the translate_v2 module and assign it to translate_client
        Call the translate method of translate_client with ontology_text and target_language as parameters and assign the output to output
        Unescape HTML entities in the 'translatedText' field of output and assign it to translated_text
        return translated_text
end function
```

Usage

The usage of the function is as follows:

```
og_language = 'it-IT'
text = "Hello, How are you?"
translator = SpeechToTextTranslator(
    project_id, language_codes, audio_file
)
```

```
text = translator.translate_text(og_language, text)
```

Translation process

(a) **Client initialization**
Using the `translate_v2.Client()`, the method initializes a Google Cloud Translation API client.

```
translate_client = translate_v2.Client()
```

(b) **Translation request**
The `translate_text` function queries the API for a translation, specifying the target language and the text to be translated.

```
output = translate_client.translate(
    ontology_text, target_language=target_language
)
```

(c) **Translated text extraction**
The translated text is extracted from the API response.

```
translated_text = html.unescape(output['translatedText'])
```

The translation may contain HTML entities, so `html.unescape` is used to convert them back to their corresponding characters.

(d) **Result return**
The translated text is returned to the calling code

```
return translated_text
```

This `translate_text` function simplifies text translation, making it appropriate for adapting the system's output to the user's preferred language or for any other context where multilingual support is required. The modular nature of this method allows for easy integration into various parts of the system that require language translation.

8. The Google text-to-speech API is used to transform the text to audio.

9. The process is repeated by playing the audio in Nao.

3.4 Interactive Storytelling

In the experiment with the children, two versions of the developed framework were used. The first framework follows a predefined storyline where the questions are asked in chronological order. The second framework is a collaborative one, where the progression of the story is influenced by the child's previous response. In this collaborative framework, the OpenAI API is used to modify the subsequent question based on the child's previous response, without altering the original intent of the question. These differences primarily occur in the section of the pipeline dealing with question preparation. Examples of both frameworks are provided below.

3.4.1 Normal Framework

A part of the normal framework is given below:

```
Nao: Hello there, little Earthlings! My name is Robo, and I come
from a faraway planet called Zogar. Where are you from, Marco?
Child: Puoi tradurlo
Nao: Ciao là, piccoli terrestri! Il mio nome è Robo e vengo da un
lontano pianeta chiamato Zogar. Da dove vieni, Luca?
Child: Vengo da Genova
Nao: Buongiorno! L'Italia è un paese bellissimo. Sapevate che la
capitale d'Italia è Roma?
Child: Si
Nao: È corretto!
Nao: A Zogar tutto brilla come diamanti e gli alberi cantano
canzoni con  il vento. Immagina che a casa mia comunichiamo
attraverso schemi di luce.
Come vi esprimete qui, Francesca?
Child: We communicate with the help of language
Nao: That's interesting! Language is indeed a powerful tool for
communication.
Nao: I have also discovered that you like eating something called
food.
In my planet, we eat crystals! What kind of food do you eat in your
country,
Francesca?
Child: We eat mostly pasta, rice, etc.
Nao: That sounds delicious! Pasta and rice are popular in many
countries.
```

3.4.2 Collaborative Framework

A part of the collaborative framework is given below:

```
Nao: Hello there, little Earthlings! My name is Robo, and I come
from a faraway planet called Zogar. Where are you from, Angela?
Child: Ich komme aus Berlin
Nao: Guten Morgen! Deutschland ist ein wunderschönes Land. Wussten
Sie, dass die Hauptstadt Deutschlands Berlin ist?
Child: Puoi tradurlo
Nao: Buongiorno! La Germania è un paese bellissimo. Sai che la
capitale della Germania è Berlino?
Child: Yes. But I have never been to Berlin.
Nao: That's interesting
Nao: On Zogar, everything sparkles like diamonds, and the trees
sing songs with the wind, just like in my home. Imagine that in my
home, we communicate through light patterns, similar to how the
vibrant
streets of Berlin captivate your senses. How do you express
yourselves
here, Angela, having never been to Berlin?
Child: We express with the help of language
Nao: That's interesting! Language is indeed a powerful tool for
expressing ourselves.
Nao: In my planet, we express our love for crystals through our
language. It's fascinating how language can help us convey our
emotions.
Speaking of expressions, I have also discovered that you like
eating
something called food. So, I'm curious, what kind of food do you
eat in
your country, Angela?
Child: I had brioche for breakfast.
Nao: That sounds delicious! Brioche is a popular choice for
breakfast in many countries.
```

Chapter 4

Evaluation

4.1 Evaluation Metrics

The evaluation metrics of this thesis are divided into two distinct aspects: technical evaluation of response times and the assessment of the generated interaction.

The technical evaluation primarily focuses on the efficiency of various components that contribute to the functionality of the social robot, Nao. This assessment includes an examination of the time required for the server socket to transmit audio to the Nao robot, the efficiency of translation services, the effectiveness of the OpenAI API, and the proficiency of the speech-to-text and text-to-speech procedures. Additionally, the system's ability to generate relevant stories within the Intercultural Pedagogy domain was evaluated through a specialized questionnaire administered to teachers.

In these interactions, it is crucial to establish a predefined upper limit of 2 seconds for the overall human-robot response time, from the moment the robot asks a question to when the child answers (Pelikan & Hofstetter, 2023). This 2-second benchmark serves as a vital measure for ensuring a seamless interaction process. By adhering to this standard, we can ensure the system's responsiveness, facilitating smooth, timely, and effective engagement with the children, thereby contributing to an optimal interactive learning environment.

The assessment of the generated interaction of the robot is evaluated based on its ability to generate relevant stories within the Intercultural Pedagogy domain. This is assessed through a specialized questionnaire administered to teachers, accompanied by a script of interaction between the robot and children from different countries.

This dual-aspect evaluation will be instrumental in identifying potential areas for development and assessing the system's existing efficiency. The ultimate goal is to enhance the social robot's efficacy and interactivity in supporting intercultural education in preschools, daycare facilities, and nurseries, thereby providing the children with an interactive and adaptable learning environment.

4.2 Technical Evaluation of Response Times

For each of these aspects, we will calculate the mean, median, and standard deviation of the 50 recorded values. These statistical measures will provide a comprehensive understanding of the performance and efficiency of each component. The mean will give us the average time taken, the median will provide the midpoint of the data, and the standard deviation will offer insight into the variability of the data. This analysis will help us identify potential areas for improvement and optimization.

4.2.1 Speech-to-Text Response Time

The speech-to-text response time refers to the duration it takes for the system to convert spoken language into written text. This process is crucial for understanding the child's responses during the interactive storytelling experiment. The efficiency of this process can significantly impact the overall user experience and the pace of the interaction.

The scatter plot (Fig. 4.1) represents the time taken for each speech-to-text task. The x-axis represents the speech-to-text number, and the y-axis represents the time taken in seconds. This plot helps us understand the efficiency of the speech-to-text process.

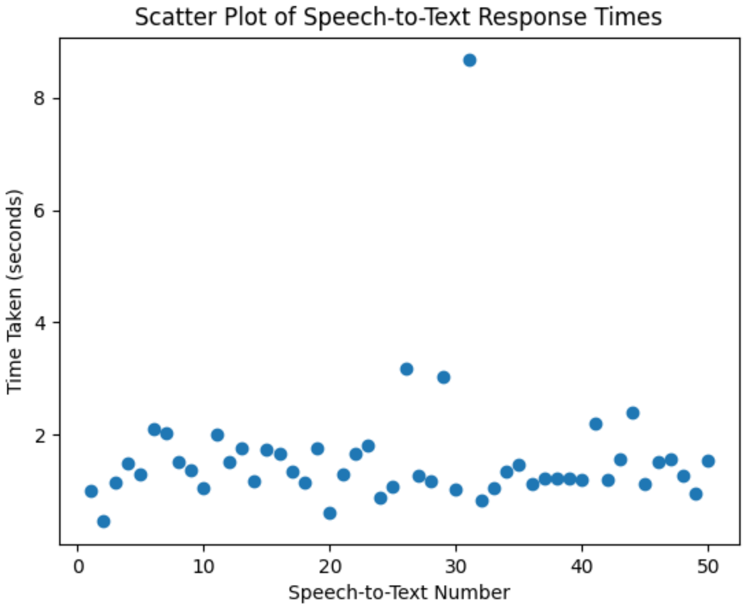

Figure 4.1: Speech-to-Text Response Times

Mean

The time values in seconds are: $1.005, 0.451, 1.135, 1.493, 1.287, \ldots, 1.541$.

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$\bar{x} = \frac{1}{50}(1.005 + 0.451 + \ldots + 1.541)$$

$$\bar{x} \approx 1.401\,\text{seconds}$$

Median

Sort the time values in ascending order: [0.452, 0.610, 0.818, 0.877, 0.951, ..., 3.182]

Since there are 50 values (an even number), the median is the average of the 25th and 26th values.

$$\text{Median} = \frac{1.223 + 1.200}{2} \approx 1.2115\,\text{seconds}$$

Standard Deviation

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$\sigma = \sqrt{\frac{1}{50}((1.005 - 1.401)^2 + (0.451 - 1.401)^2 + \ldots + (1.541 - 1.401)^2)}$$

$$\sigma \approx 0.669\,\text{seconds}$$

So, the calculated values are:
Mean ($\bar{x}$): $\approx 1.401 \approx 1.401$ seconds
Median: $\approx 1.2115 \approx 1.2115$ seconds
Standard Deviation (σ): $\approx 0.669 \approx 0.669$ seconds

4.2.2 Text-to-Speech Response Time

The text-to-speech response time is the time taken by the system to convert written text into spoken language. This is an essential aspect of the system as it allows the robot to communicate with the children using audible speech. The speed of this process can affect the fluidity of the conversation with the robot.

The time taken for each text-to-speech task is represented in the scatter plot (Fig. 4.2). Similar to the first plot, the x-axis represents the text-to-speech number, and the y-axis represents the time taken in seconds. This plot provides insight into the performance of the text-to-speech process.

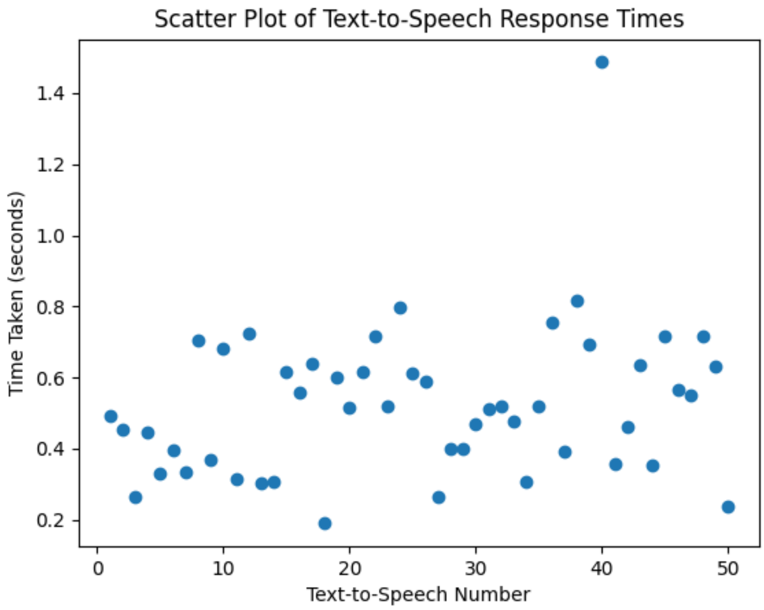

Figure 4.2: Text-to-Speech Response Times

Mean

The time values in seconds are: $0.492, 0.455, 0.263, 0.447, 0.330, \ldots, 0.308$.

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

$$\bar{x} = \frac{1}{50}(0.492 + 0.455 + \ldots + 0.308)$$

$$\bar{x} \approx 0.4929 \, \text{seconds}$$

Median

Sort the time values in ascending order: [0.189, 0.264, 0.263, 0.303, 0.306, ..., 0.817]

Since there are 50 values (an even number), the median is the average of the 25th and 26th values.

$$\text{Median} = \frac{0.400 + 0.401}{2} \approx 0.4005 \, \text{seconds}$$

Standard Deviation

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$\sigma = \sqrt{\frac{1}{50}((0.492 - 0.4929)^2 + (0.455 - 0.4929)^2 + \ldots + (0.308 - 0.4929)^2)}$$

$$\sigma \approx 0.1488\,\text{seconds}$$

So, the calculated values are:
Mean ($\bar{x}$): $\approx 0.4929 \approx 0.4929$ seconds
Median: $\approx 0.4005 \approx 0.4005$ seconds
Standard Deviation (σ): $\approx 0.1488 \approx 0.1488$ seconds

4.2.3 Translation Response Time

The translation response time measures the duration it takes for the system to translate text from one language to another. In the context of this study, it is used to translate the robot's questions into the child's language and vice versa. The efficiency of the translation process is vital for ensuring clear and accurate communication between the robot and the child.

The time taken for each translation task is represented by the scatter plot (Fig. 4.3). The x-axis represents the translation number, and the y-axis represents the time taken in seconds. The plot shows the variation in translation times across the 50 iterations.

Mean

The time values in seconds are: $0.309, 0.249, 0.252, 0.231, 0.231, \ldots, 0.308$.

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$\bar{x} = \frac{1}{50}(0.309 + 0.249 + \ldots + 0.308)$$

$$\bar{x} \approx 0.2709\,\text{seconds}$$

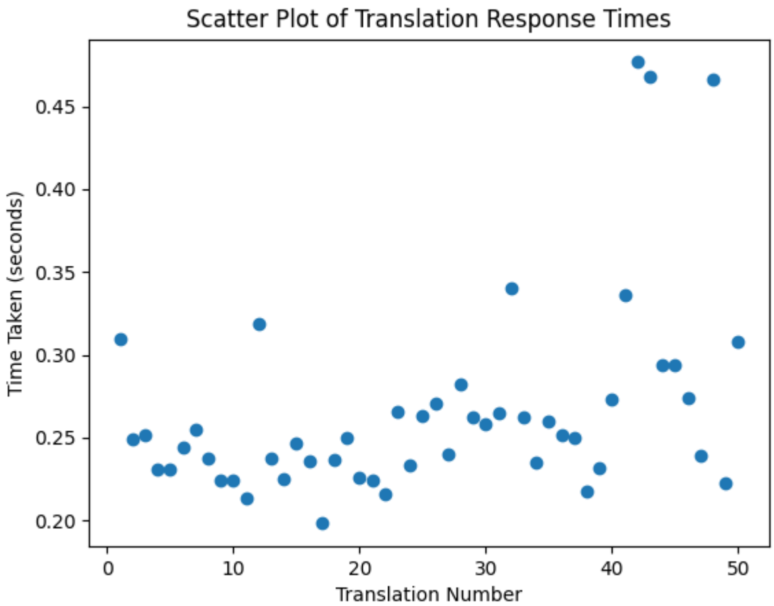

Figure 4.3: Translation Response Times

Median

Sort the time values in ascending order: [0.198, 0.213, 0.216, 0.217, 0.222, 0.223, . . . , 0.477]

Since there are 50 values (an even number), the median is the average of the 25th and 26th values.

$$\text{Median} = \frac{0.240 + 0.236}{2} \approx 0.238\,\text{seconds}$$

Standard Deviation

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$\sigma = \sqrt{\frac{1}{50}((0.309 - 0.2709)^2 + (0.249 - 0.2709)^2 + \ldots + (0.308 - 0.2709)^2)}$$

$$\sigma \approx 0.0522\,\text{seconds}$$

So, the calculated values are:

Mean ($\bar{x}$): $\approx 0.2709 \approx 0.2709$ seconds
Median: $\approx 0.238 \approx 0.238$ seconds
Standard Deviation (σ): $\approx 0.0522 \approx 0.0522$ seconds

4.2.4 OpenAI API Response Time

The OpenAI API response time refers to the time taken for the system to interact with the OpenAI API. This includes the time taken to send a request to the API and receive a response. The OpenAI API is used for several tasks in the system, including understanding the child's responses and generating appropriate questions. The response time of the API can influence the overall performance of the system.

The time taken for each OpenAI API request by the scatter plot (Fig. 4.4). The x-axis represents the OpenAI API request number, and the y-axis represents the time taken in seconds. This plot provides a measure of the performance of the OpenAI API.

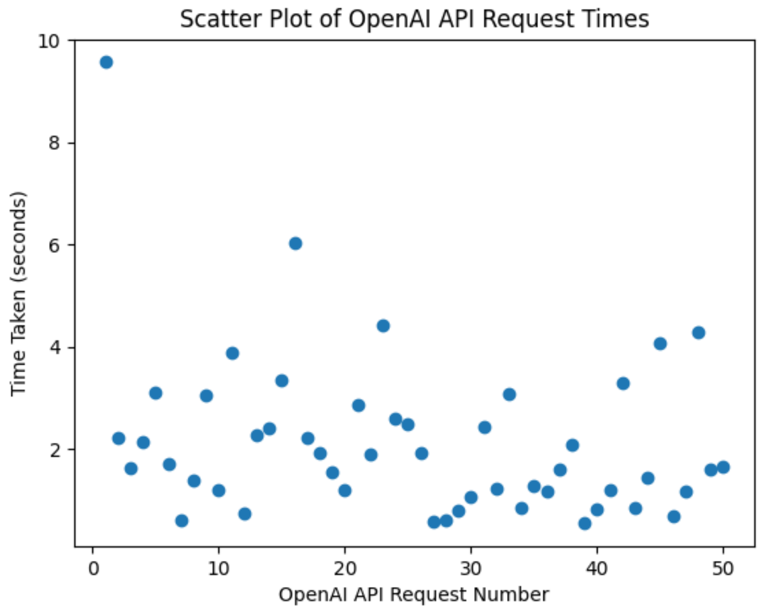

Figure 4.4: OpenAI API Response Times

Mean

The time values in seconds are: $9.571, 2.233, 1.641, 2.135, 3.110, \ldots, 1.664$.

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$\bar{x} = \frac{1}{50}(9.571 + 2.233 + \ldots + 1.664)$$

$$\bar{x} \approx 1.802\,\text{seconds}$$

Median

Sort the time values in ascending order: [0.549, 0.594, 0.605, 0.608, 0.699, ..., 9.571]

Since there are 50 values (an even number), the median is the average of the 25th and 26th values.

$$\text{Median} = \frac{1.288 + 1.183}{2} \approx 1.236\,\text{seconds}$$

Standard Deviation

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$\sigma = \sqrt{\frac{1}{50}((9.571 - 1.802)^2 + (2.233 - 1.802)^2 + \ldots + (1.664 - 1.802)^2)}$$

$$\sigma \approx 1.387\,\text{seconds}$$

So, the calculated values are:
Mean ($\bar{x}$): $\approx 1.802 \approx 1.802$ seconds
Median: $\approx 1.236 \approx 1.236$ seconds
Standard Deviation (σ): $\approx 1.387 \approx 1.387$ seconds

4.2.5 Audio Transmission Time to Nao

The audio transmission time to Nao measures the duration it takes to send audio data to the Nao robot. This is crucial for assessing the efficiency of the communication process involving audio.

The time taken to send audio to Nao is represented by the scatter plot(Fig. 4.5). The x-axis represents the audio sending number, and the y-axis represents the time taken in seconds. This plot gives us an understanding of the efficiency of the audio transmission process to Nao.

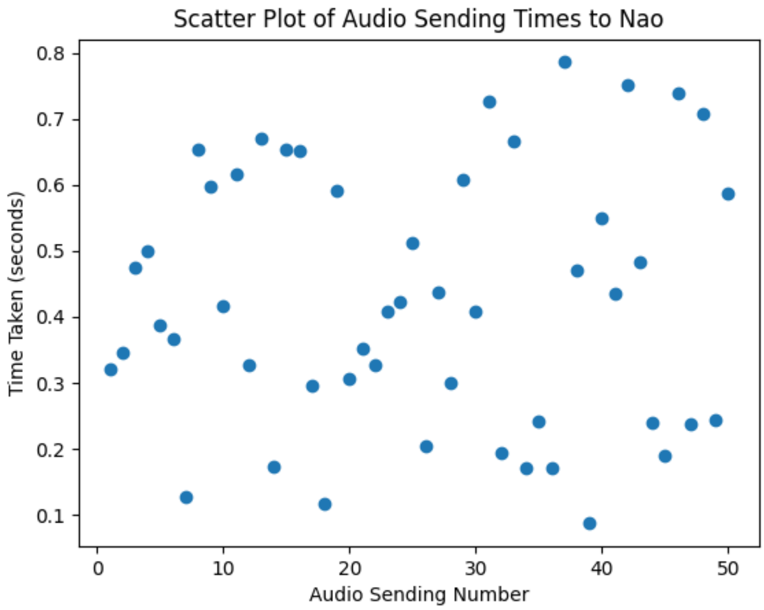

Figure 4.5: Audio Transmission Response Times

Mean

The time values in seconds are: $0.321, 0.346, 0.475, 0.499, 0.387, \ldots, 0.587$.

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$\bar{x} = \frac{1}{50}(0.321 + 0.346 + \ldots + 0.587)$$

$$\bar{x} \approx 0.424\,\text{seconds}$$

Median

Sort the time values in ascending order: [0.087, 0.117, 0.127, 0.172, 0.172, ..., 0.786]

Since there are 50 values (an even number), the median is the average of the 25th and 26th values.

$$\text{Median} = \frac{0.408 + 0.423}{2} \approx 0.415\,\text{seconds}$$

Standard Deviation

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$\sigma = \sqrt{\frac{1}{50}((0.321 - 0.424)^2 + (0.346 - 0.424)^2 + \ldots + (0.587 - 0.424)^2)}$$

$$\sigma \approx 0.151\,\text{seconds}$$

So, the calculated values are:
Mean ($\bar{x}$): $\approx 0.424 \approx 0.424$ seconds
Median: $\approx 0.415 \approx 0.415$ seconds
Standard Deviation (σ): $\approx 0.151 \approx 0.151$ seconds

4.2.6 Comparison of Response times

In this section, the individual response times for each component of the system, including Speech-to-Text (STT), Text-to-Speech (TTS), Translation, OpenAI API, and Audio Transmission to Nao, are presented. These response times serve as critical metrics in evaluating the efficiency of each module. Below are the mean response times for each component:

- Speech-to-Text: 1.401 seconds
- Text-to-Speech: 0.49 seconds
- Translation: 0.270 seconds
- OpenAI API: 1.802 seconds
- Audio Transmission to Nao: 0.424 seconds

Considering the overall response time, the duration from the child's response to the subsequent question can be calculated by summing up the average individual response times. Therefore, the average overall response time is approximately 4.287 seconds, exceeding the predefined upper limit of 2 seconds.

Analyzing the contributions of individual components to the overall response time helps identify critical areas for improvement. In this case, the OpenAI API exhibits the highest mean response time, indicating a potential bottleneck. Addressing this bottleneck could significantly enhance the overall system responsiveness. Strategies such as optimizing API prompts, using a lower model than GPT-3.5 Turbo, reducing the number of tokens, parallel processing, or exploring alternative solutions may be considered to diminish the OpenAI API response time. It is also observed that OpenAI servers occasionally experience substantial server load, leading to increased response times. Additionally, Google Cloud Speech-to-Text also incurs a significant time for response. The response time of the Speech-to-Text API can be enhanced by using shorter audio segments, adjusting configuration parameters, or employing more enhanced specialized language models.

The comparison of mean, median, and Standard deviation for all the response times are given represented in the figures 4.6, 4.7, and 4.8.

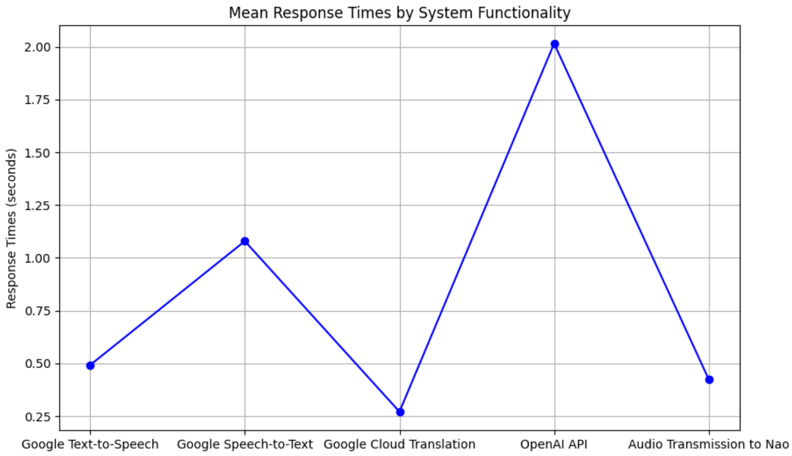

Figure 4.6: Mean Response Times

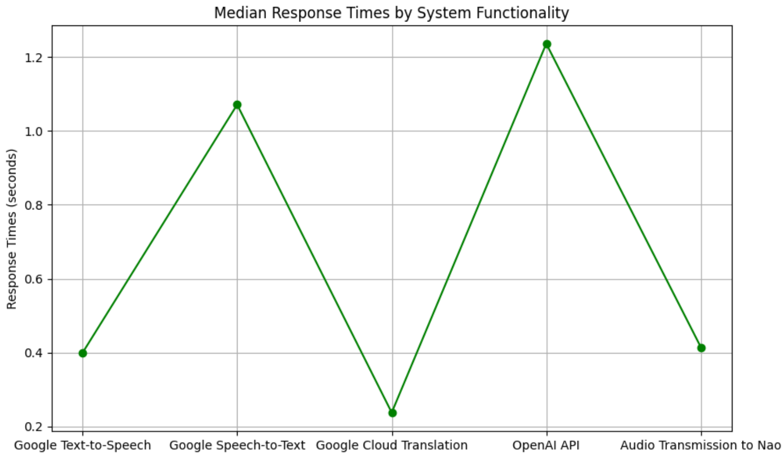

Figure 4.7: Median Response Times

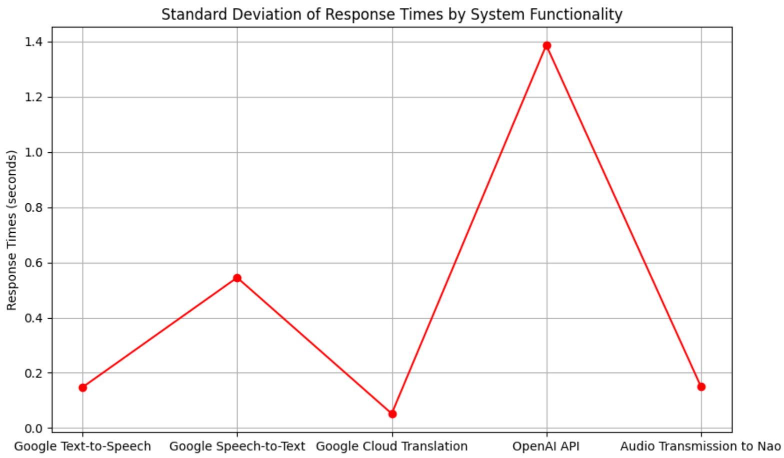

Figure 4.8: Standard Deviation of Response Times

4.3 Generated Interaction Assessment

In the evaluation phase of the thesis project, an online questionnaire along with a script of interaction between the robot and children from different countries(A.2) was administered to the teachers of a kindergarten. This questionnaire aimed at studying techniques to promote integration and inclusion of children in schools through the use of humanoid robots.

The questionnaire consisted of six multiple-choice questions and two open-ended questions (which were not mandatory). The teachers were asked to read a brief text representing a potential verbal interaction between a social robot and a group of three children aged between 4 and 6 years, potentially from different countries. In the example provided, one child was German, while the other two were Italian. The interaction was completely autonomous, meaning that everything the robot said was decided by the controlling software, and the robot was not teleoperated.

The interaction script presented in the questionnaire was designed to mimic a real-life scenario where the robot engages in a culturally relevant dialogue with the children, asking questions and responding in their native languages. This script served as a basis for the teachers to evaluate the effectiveness and relevance of such an interaction in a multicultural educational setting.

No personal data was collected during this process, and the responses to the questionnaire were completely anonymous. The feedback gathered from this questionnaire was invaluable in assessing the practical implications and effectiveness of using social robots for intercultural pedagogy in early childhood education settings. The insights gained from this evaluation process have significantly contributed to the findings and conclusions of the thesis project.

The questionnaire utilized for the evaluation process can be found in Appendix A.2.

4.3.1 Evaluation of Questionnaire

The evaluation of the questionnaire was conducted by calculating descriptive statistics such as the mean, median, mode, and standard deviation for each statement in the questionnaire. The results are as follows:

First Question

Q. If a small humanoid robot were used in a classroom and interacted with children in this way, the interaction may have an effect on the inclusion of children who do not speak Italian well (or at all)

The values given by the teachers are represented by a scatter plot as shown in Fig. 4.9

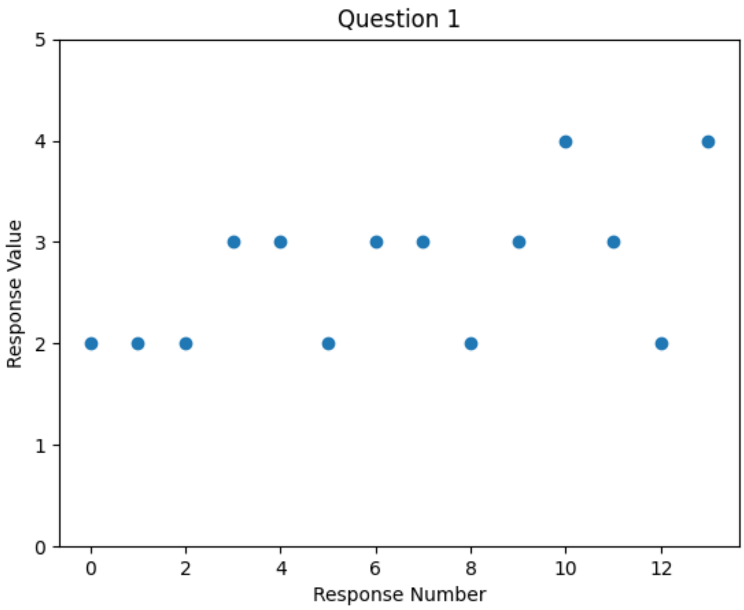

Figure 4.9: Responses of Question 1

1. **Mean**: (2+2+2+3+3+2+3+3+2+3+4+3+2+4)/14 = 2.71
2. **Median**: The sorted list is 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 4, 4. The median is (3+3)/2 = 3.
3. **Mode**: The number 3 appears most frequently, so the mode is 3.
4. **Standard Deviation**: First, calculate the variance. The variance is the average of the squared differences from the mean. Then, take the square root of the variance to get the standard deviation. The standard deviation is approximately 0.70.

Based on these calculations, we can conclude that the majority of responses are around the value of 3, indicating that most respondents agree or strongly agree with the statements in the questionnaire. The standard deviation of 0.70 shows that the responses are not widely spread out, suggesting a general consensus among the respondents. This could indicate that the use of a humanoid robot in a classroom may indeed have a positive effect on the inclusion of children who do not speak Italian well or at all.

Second Question

Q. The robot treats all children with respect.

The values given by the teachers are represented by a scatter plot as shown in Fig. 4.10

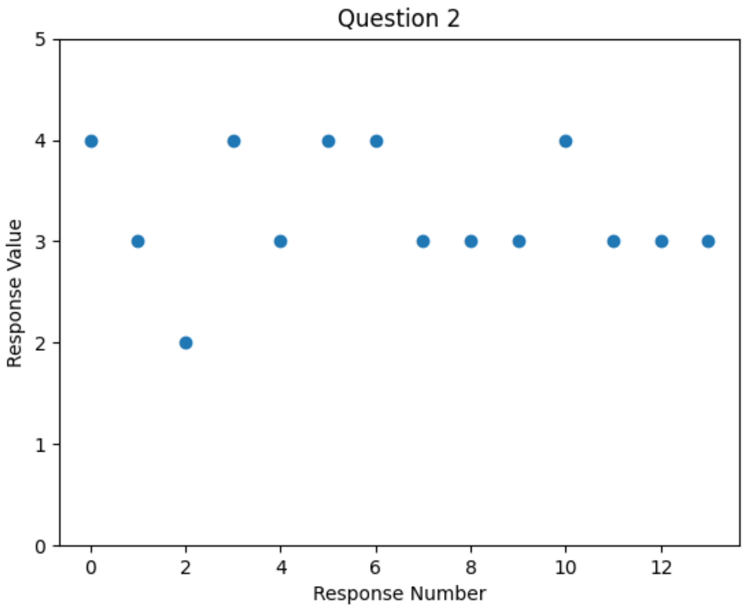

Figure 4.10: Responses of Question 2

1. **Mean**: (4+3+2+4+3+4+4+3+3+3+4+3+3+3)/14 = 3.36
2. **Median**: The sorted list is 2, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4. The median is (3+3)/2 = 3.
3. **Mode**: The number 3 appears most frequently, so the mode is 3.
4. **Standard Deviation**: First, calculate the variance. The variance is the average of the squared differences from the mean. Then, take the square root of the variance to get the standard deviation. The standard deviation is approximately 0.63.

Based on these calculations, we can conclude that the majority of responses are around the value of 3, indicating that most respondents agree or strongly agree with the statement in the questionnaire. The standard deviation of 0.63 shows that the responses are not widely spread out, suggesting a general consensus among the respondents. This could indicate that the robot is perceived to treat all children with respect.

Third Question

Q. The robot seems aware of children's cultural differences.

The values given by the teachers are represented by a scatter plot as shown in Fig. 4.11

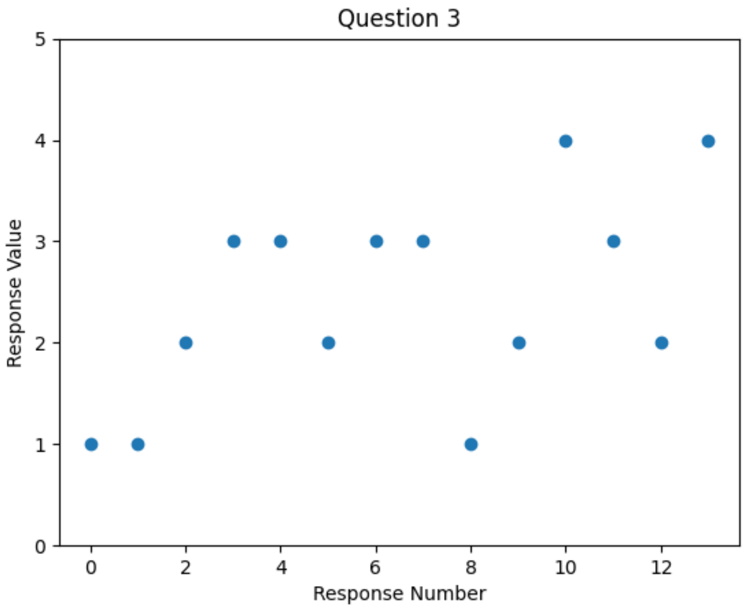

Figure 4.11: Responses of Question 3

1. **Mean**: (1+1+2+3+3+2+3+3+1+2+4+3+2+3)/14 = 2.36
2. **Median**: The sorted list is 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 4. The median is (2+3)/2 = 2.5.
3. **Mode**: The number 3 appears most frequently, so the mode is 3.
4. **Standard Deviation**: First, calculate the variance. The variance is the average of the squared differences from the mean. Then, take the square root of the variance to get the standard deviation. The standard deviation is approximately 0.85

Based on these calculations, we can conclude that the majority of responses are around the value of 2 or 3, indicating that most respondents are neutral or agree with the statement in the questionnaire. The standard deviation of 0.85 shows that the responses are somewhat spread out, suggesting a variety of opinions among the respondents. This could indicate that while some respondents believe the robot is aware of the children's cultural differences, others may not share this view.

Fourth Question

Q. If a small humanoid robot were used in a classroom and interacted with children in this way, the interaction could encourage the inclusion of children from different countries.

The values given by the teachers are represented by a scatter plot as shown in Fig. 4.12

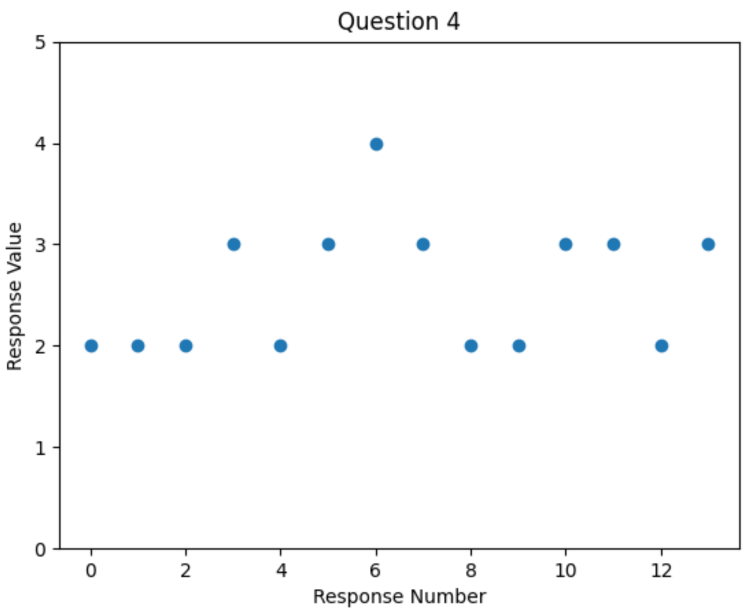

Figure 4.12: Responses of Question 4

1. **Mean**: (2+2+2+3+2+3+4+3+2+2+3+3+2+3)/14 = 2.64
2. **Median**: The sorted list is 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 4. The median is (2+3)/2 = 2.5.
3. **Mode**: The number 2 appears most frequently, so the mode is 2.
4. **Standard Deviation**: First, calculate the variance. The variance is the average of the squared differences from the mean. Then, take the square root of the variance to get the standard deviation. The standard deviation is approximately 0.63.

Based on these calculations, we can conclude that the majority of responses are around the value of 2 or 3, indicating that most respondents are neutral or agree with the statement in the questionnaire. The standard deviation of 0.63 shows that the responses are not widely spread out, suggesting a variety of opinions among the respondents. This could indicate that while some respondents believe the robot may encourage the inclusion of children from different countries, others may not share this view.

Fifth Question

Q. A robot with a tablet (on which to show images related to the story) could be much more effective in this context.

The values given by the teachers are represented by a scatter plot as shown in Fig. 4.13

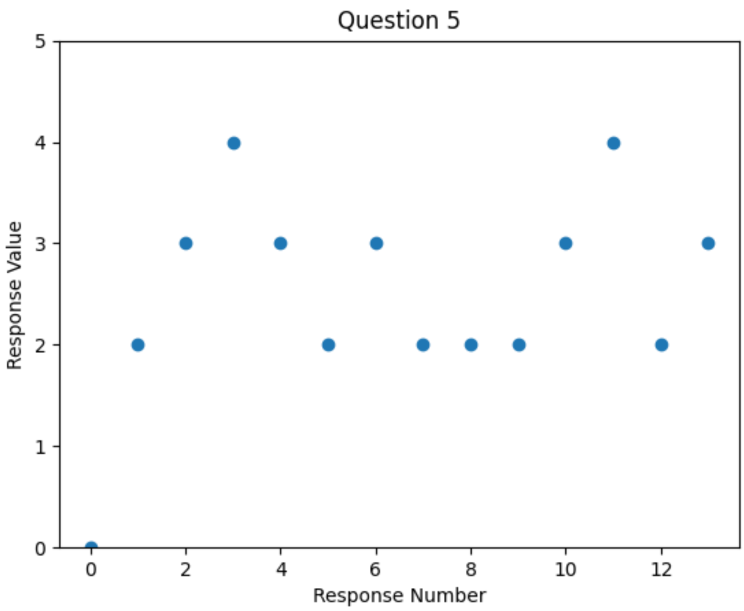

Figure 4.13: Responses of Question 5

1. **Mean**: (0+2+3+4+3+2+3+2+2+2+3+4+2+3)/14 = 2.57
2. **Median**: The sorted list is 0, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 4. The median is (2+3)/2 = 2.5.
3. **Mode**: The number 2 appears most frequently, so the mode is 2.

4. **Standard Deviation**: First, calculate the variance. The variance is the average of the squared differences from the mean. Then, take the square root of the variance to get the standard deviation. The standard deviation is approximately 0.85.

Based on these calculations, we can conclude that the majority of responses are around the value of 2 or 3, indicating that most respondents are neutral or agree with the statement in the questionnaire. The standard deviation of 0.85 shows that the responses are somewhat spread out, suggesting a variety of opinions among the respondents. This could indicate that while some respondents believe a robot with a tablet could be more effective in this context, others may not share this view.

Sixth Question

Q. I wouldn't want a robot like that in the classroom

The values given by the teachers are represented by a scatter plot as shown in Fig. 4.14

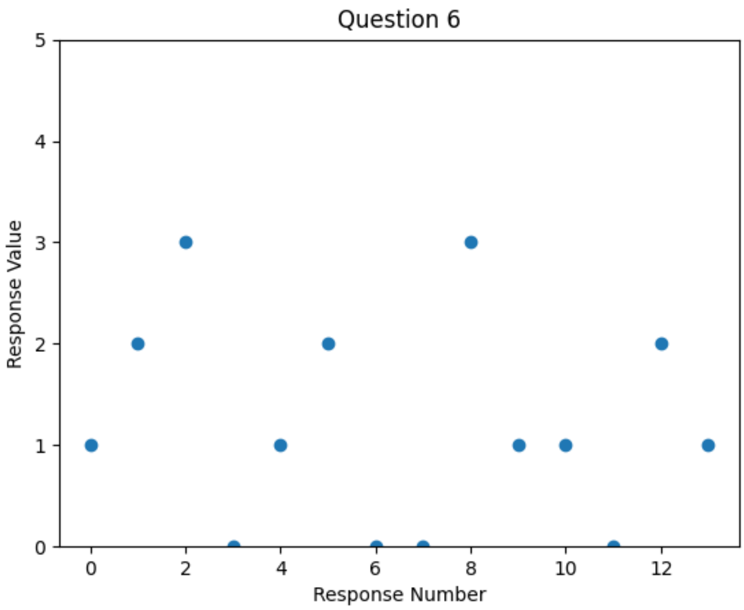

Figure 4.14: Responses of Question 6

1. **Mean (Average)**: The mean is calculated by adding up all the numbers and then dividing by the count of numbers.

2. **Median**: The median is the middle number in a sorted list of numbers. If the list has an even number of observations, the median is the average of the two middle numbers.

3. **Mode**: The mode is the number that appears most frequently in a data set.

4. **Standard Deviation**: The standard deviation is a measure of how spread out numbers are. It is the square root of the variance.

Based on these calculations, we can conclude that the majority of responses are around the value of 1, indicating that most respondents are neutral or disagree with the statement in the questionnaire. The standard deviation of 0.83 shows that the responses are somewhat spread out, suggesting a variety of opinions among the respondents. This could indicate that while some respondents may not want a robot like that in the classroom, others may not share this view.

Teachers were also asked some optional descriptive questions, to which most of them did not respond. One of the teachers inquired about what the robot would do if the children did not listen to it. Additionally, some teachers suggested conducting fun activities to prevent the children from getting bored.

The outcomes of the six questions suggest that a significant portion of the participants had a neutral to positive outlook on the deployment of a humanoid robot in a classroom environment. A majority concurred that the robot exhibits respect towards all children and demonstrates an understanding of their cultural diversities. They also believed that the interaction between the robot and the children could encourage the inclusion of children from different countries and those who do not speak Italian well or at all.

A critical finding from the evaluation emerged when respondents were asked whether a robot equipped with a tablet could be more effective than using the robot alone. The responses revealed a diversity of opinions. Some respondents suggested that a robot with a tablet could be more effective in this context, while others expressed reservations about having such a robot in the classroom. A potential solution could involve initiating a pilot program where a robot with a tablet is introduced into a select number of classrooms. This approach would facilitate real-world testing and the collection of feedback. Based on these results and feedback, necessary modifications could be implemented before a wider rollout is considered. This strategy ensures a careful and measured approach to integrating technology into classrooms.

These findings highlight the potential benefits and challenges of using social robots in educational settings, particularly in promoting cultural and linguistic diversity. However, further research and more data are needed to confirm these conclusions and to explore additional ways to enhance the effectiveness of social robots in multicultural classrooms. This could include incorporating fun activities, as suggested by some teachers, to engage the children and maintain their interest.

Chapter 5

Conclusions

5.1 Research Review

This research aimed at creating a social robot that helps to promote intercultural pedagogy among young children. The first main objective of the thesis was to develop an application that helps in transcribing speech to text, converting text to speech, using translation services to translate sentences from one language to another, processing information from the text, and retrieving information from a knowledge database using Google Cloud API, OpenAI API, and Protégé database. The other important objective was to connect the social robot Nao to the system using the Naoqi module and integrate it with the collaboration of Google Cloud API, OpenAI API, and the Protégé knowledge database.

The technical evaluation of response times across various system functionalities offers crucial insights into the efficiency and responsiveness of the implemented components. The translation process exhibits rapid and consistent performance, emphasizing its reliability. Similarly, Text-to-Speech functionality demonstrates commendable speed. The audio transmission time to Nao is also consistent in terms of performance.

However, it's essential to highlight that OpenAI API requests and Google Cloud Speech-to-Text responses contribute significantly to the overall delay. Identifying these components as potential bottlenecks indicates that improvements in these areas would substantially enhance the overall system performance. Importantly, the current implementation experiences a delay that exceeds the acceptable limits of 2 seconds of overall response time. Therefore, future efforts will be directed toward optimizing these components and reducing the overall time required for the system to respond to user inputs.

In the generated interaction phase of the thesis project, an online questionnaire along with a script of interaction between the robot and the child was administered to kindergarten teachers to study the use of humanoid robots for promoting integration and inclusion in schools. The questionnaire, comprising six multiple-choice and two optional open-ended questions, required teachers to review the script of a potential autonomous interaction between a social robot and children of different nationalities. The interaction script, designed to mimic a real-life scenario, involved the robot engaging in culturally relevant dialogues in the children's native languages. The responses of the respondents were evaluated in terms of mean, median, mode, and standard deviation. It was found that the majority of the respondents had a neutral to positive attitude towards the use of a humanoid robot in a classroom setting. The anonymous feedback from the questionnaire provided valuable insights into the practical implications and effectiveness of using social robots for intercultural pedagogy in early childhood education, significantly contributing to the thesis project's findings and conclusions.

5.2 Limitations

During the execution of the project, several key challenges were identified. One of the main issues was the limitation in language support for the speech-to-text operation, which currently only supports three languages. This restricts the system's ability to interact with children from a wider range of cultural backgrounds.

One of the main issues was that the system did not function optimally in noisy environments. This is a crucial aspect to address, as the target users are children who often interact in lively and noisy settings. Enhancements in noise filtering and speech recognition algorithms could be explored to improve the system's performance in such environments.

Another challenge was the inaccurate responses from the OpenAI API during the collaborative storytelling session. This could potentially disrupt the flow of the story and lead to confusion among the children. Future work could focus on refining the interaction with the OpenAI API to ensure more accurate and contextually appropriate responses.

Another significant challenge was the lengthy response times from the OpenAI API, which could potentially disrupt the flow of the storytelling session and reduce the engagement of the children. Furthermore, there were instances where the OpenAI API provided inaccurate responses during the collaborative storytelling session, which could lead to confusion or misunderstanding.

Issues were also encountered with the Google Cloud APIs. Occasionally, languages other than English were mistakenly identified as English during the speech-to-text operation. This could lead to incorrect processing of the children's responses. Moreover, the system struggled to accurately detect speech in noisy environments, which is a common scenario in group settings with children.

It's also important to note that due to time constraints, the system has not yet been tested in a real-world environment. As such, the system's ability to accurately detect and process the children's speech in such settings remains an open question.

In addition, the quality of internet connectivity in the experimental workspace is a crucial factor for the smooth operation of the system. Poor connectivity could lead to delays or disruptions in the storytelling session.

Moving forward, addressing these challenges will be a key focus. Enhancements could include expanding language support, optimizing API response times, improving speech recognition accuracy, and conducting real-world testing. These improvements would significantly enhance the system's effectiveness and usability, making it a more powerful tool for promoting intercultural understanding among children.

5.3 Future Work

Further development could greatly improve the storytelling application by increasing the knowledge base to include more cultures and more in-depth details about each culture by adding more data to the database. This would enable the robot to give the kids a more comprehensive and varied cultural learning experience. Furthermore, adding gestures to the robot's speech could improve the storytelling and children's interaction with it by making it more realistic and engaging.

To widen its application, the system might be modified to function with other types of robots or virtual avatars. Incorporating machine learning techniques to allow the robot to learn from every interaction and gradually enhance its narrative and question-asking skills might be an intriguing new area for future work. This could improve the system's effectiveness and engagement in helping kids develop multicultural awareness.

Moreover, the system's efficiency might be increased by using Choregraphe to integrate the application directly within the Nao. Finally, it is crucial to conduct real-world testing in environments like schools and after-school activities. This would present opportunities for additional development and enhancement as well as insightful information on the system's efficacy in a real-world setting. These improvements and additions may greatly boost the application's ability to foster children's intercultural understanding.

Appendix A

Extra

A.1 Protégé Database

The Protege database, detailed in this appendix, is a crucial component of this study. It is structured with a hierarchy of subclasses and object characteristics, all falling under the primary class of owl:Thing. This hierarchical structure allows for a comprehensive and organized representation of the data. The database covers a broad range of topics, including geographical regions, cultural features, languages, and time of day. It is flexible enough to include data from various continents and nations, with a particular focus on Europe, especially Italy. The ontology makes use of several object properties, each with its domains and ranges, to establish meaningful relationships between different classes and subclasses. Each subclass has numerous direct instances, and each of these instances encompasses multiple property assertions. This appendix provides a detailed overview of the class hierarchy, object properties, direct instances, and object property assertions in the Protege database. It serves as a reference for understanding the structure and content of the database used in this study.

A.1.1 Class Hierarchy

A.1.1.1 Geographical Hierarchy

The geographical hierarchy is divided into two main subclasses: continents and countries.

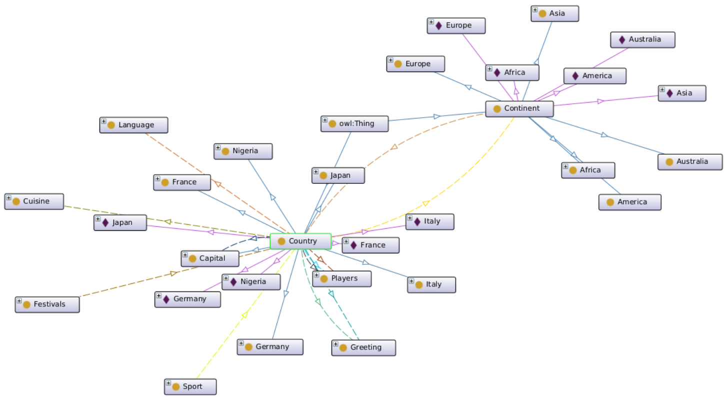

Figure A.1: Geographical Hierarchy

- The **Continents** subclass includes five continents:
  - Africe
  - America
  - Asia
  - Australia
  - Europe
- The **Countries** subclass expands into:
  - Capital: This subclass denotes the capitals of various countries.
  - Countries: Specific data for five countries are included:
    * France
    * Germany
    * Italy
    * Japan
    * Nigeria

A.1.1.2 Cultural Hierarchy

The cultural hierarchy is categorized into five subclasses: Cuisine, DishDescription, Festivals, Greeting, and Sport.

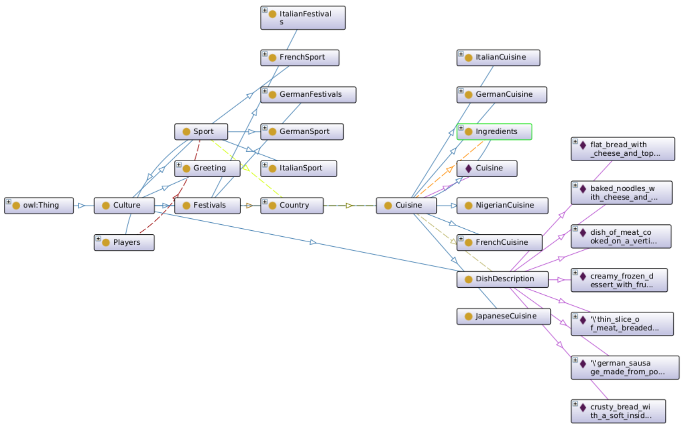

Figure A.2: Cultural Hierarchy

- **Cuisine**: This subclass includes:
  - **Ingredients**: This subclass lists various ingredients used in different cuisines.
  - **Country-Specific Cuisines**: There are subclasses for specific cuisines from different countries.
    - ∗ ItalianCuisine
    - ∗ GermanCuisine
    - ∗ FrenchCuisine
    - ∗ JapaneseCuisine
    - ∗ NigerianCuisine
- **Festivals**: This subclass lists various Festivals in various countries. This subclass includes:
  - ItalianFestivals
  - GermanFestivals
- **Greetings**: This subclass lists various greetings used in different countries. This subclass includes:
  - FrenchGreeting
  - GermanGreeting
  - ItalianGreeting

  - JapaneseGreeting
  - NigerianGreeting

- **Sport**: This subclass lists various sports used in different countries. This subclass includes:
  - FrenchSport
  - GermanSport
  - ItalianSport
  - **Players**: This subclass lists the names of various players.

The Protégé database's detailed hierarchical structure makes it possible to present the data in an extensive and well-organized manner, which makes it easier to retrieve and analyze the data for the study. It offers a strong framework for the research with the adaptability to add more information as needed.

A.1.2 Object Properties

This study's ontology makes use of a number of object properties, each with their own domains and ranges. These properties help the ontology's relationships between different classes and subclasses.

Here is a detailed breakdown of the object properties:

- **hasCountry**: This property has a domain of Continent and a range of Country.
- **hasFood**: This property has a domain of Country and a range of Cuisine.
- **hasContinent**: This property has a domain of Country and a range of Continent.
- **hasDescription**: This property has a domain of Cuisine and a range of DishDescription.
- **hasFootballPlayers**: This property has a domain of Country and a range of Players.
- **hasGreeting**: This property has a domain of Country and a range of Greeting.
- **hasIngredient**: This property has a domain of Cuisine and a range of Ingredients.

- **hasLanguage**: This property has a domain of Country and a range of Language.
- **hasTimeOfDay**: This property has a domain of Country and a range of TimeOfDay.
- **hasTranslation**: This property has a domain of Sentences and a range of Translation. It has two subproperties:
  - **hasGermanTranslation**: This subproperty has a domain of Sentences and a range of GermanTranslation.
  - **hasItalianTranslation**: This subproperty has a domain of Sentences and a range of ItalianTranslation.
- **hasVolleyballPlayers**: This property has a domain of Country and a range of Players.
- **isAFestivalOf**: This property has a domain of Festivals and a range of Country.
- **isAPlayerOf**: This property has a domain of Players and a range of Sport.
- **isPopularIn**: This property has a domain of Sport and a range of Country.

These object attributes serve as the ontology's foundation, allowing meaningful relationships to be established between distinct classes and subclasses. This structure improves the database's comprehensiveness and flexibility, making it a reliable instrument for data retrieval and analysis in this study.

A.1.3 Direct Instances and Object Property Assertions

Each subclass has numerous direct instances, and each of these instances encompasses multiple property assertions.

Country Subclass

The Country subclass encompasses instances representing various countries, including France, Germany, Italy, Japan, and Nigeria. Each country instance is associated with relevant information through object property assertions:

- **France**:
  - hasCapital some Paris

  - hasContinent some Europe

- **Germany**:
  - hasLanguage some German
  - hasCapital some Berlin
  - hasContinent some Europe

- **Italy**:
  - hasLanguage some Italian
  - hasCapital some Rome
  - hasContinent some Europe

- **Japan**:
  - hasContinent some Asia

- **Nigeria**:
  - hasContinent some Africa

Cuisine Subclass

Distinct subclasses represent French, German, and Italian cuisines, each with specific dishes and ingredients:

- **French Cuisine**
  - **Croissant**

- **German Cuisine**
  1. **Bratwurst**
     - Object Property Assertions:
       * hasIngredient some Pork
       * hasIngredient some Veal
       * hasFood some Germany
       * hasDescription some 'german_sausage_made_from_pork,_beef,_or_veal'
  2. **Brotchen**
     - Object Property Assertions:

        * hasIngredient some Yeast
        * hasIngredient some Flour
        * hasIngredient some Sugar
        * hasFood some Germany
        * hasDescription some 'crusty_bread_with_a_soft_inside.'

    3. **Doner**
        - Object Property Assertions:
            * hasIngredient some Ground_Beef
            * hasIngredient some Bread
            * hasIngredient some Vegetables
            * hasFood some Germany
            * hasDescription some 'dish_of_meat_cooked_on_a_vertical_rotisserie.'
    4. **Schnitzel**
        - Object Property Assertions:
            * hasIngredient some Bread_Crumps
            * hasIngredient some Chicken
            * hasIngredient some Flour
            * hasFood some Germany
            * hasDescription some 'thin_slice_of_meat,_breaded,_and_fried'

- **Italian Cuisine**
    1. **Gelato**
        - Object Property Assertions:
            * hasIngredient some Sugar
            * hasIngredient some Milk
            * hasIngredient some Cream
            * hasFood some Italy
            * hasDescription some 'creamy_frozen_dessert_with_fruit_flavours'
    2. **Pizza**
        - Object Property Assertions:
            * hasIngredient some Tomato
            * hasIngredient some Flour
            * hasIngredient some Mozzarella_Cheese
            * hasIngredient Olive_Oil

    * hasFood some Italy
    * hasDescription some 'flat_bread_with_cheese_and_toppings'

3. **Lasagna**
   - Object Property Assertions:
     * hasIngredient some Ground_Beef
     * hasIngredient some Lasagna_Noodles
     * hasIngredient some Tomato
     * hasFood some Italy
     * hasDescription some 'baked_noodles_with_cheese_and_sauce'

Instances under DishDescription represent descriptions of various dishes:

- 'german_sausage_made_from_pork,_beef,_or_veal'
- 'thin_slice_of_meat,_breaded,_and_fried'
- 'dish_of_meat_cooked_on_a_vertical_rotisserie.'
- 'crusty_bread_with_a_soft_inside.'
- 'baked_noodles_with_cheese_and_sauce'
- 'flat_bread_with_cheese_and_toppings'
- 'creamy_frozen_dessert_with_fruit_flavours'

Festival Subclass

Instances within the GermanFestivals and ItalianFestivals subclasses represent festivals celebrated in Germany and Italy, respectively.

- **German Festivals**
  1. **Christmas**
     - **Object Property Assertions**:
       * isAFestivalOf some Germany
  2. **Karneval**
     - **Object Property Assertions**:
       * isAFestivalOf some Germany
  3. **MaiFest**

  - **Object Property Assertions**:
    * isAFestivalOf some Germany

4. **OktoberFest**
  - **Object Property Assertions**:
    * isAFestivalOf some Germany

- **Italian Festivals**

1. **Battle_Of_Oranges**
  - **Object Property Assertions**:
    * isAFestivalOf some Italy
2. **Carnevale**
  - **Object Property Assertions**:
    * isAFestivalOf some Italy
3. **Festa_della_Madonna_Bruna**
  - **Object Property Assertions**:
    * isAFestivalOf some Italy
4. **Marriage_of_the_Sea**
  - **Object Property Assertions**:
    * isAFestivalOf some Italy
5. **Regatta**
  - **Object Property Assertions**:
    * isAFestivalOf some Italy

Greetings Subclass

Subclasses delineate greetings in French, German, and Italian, each associated with specific times of the day:

- **French Greeting**

1. **Bonjour**:
  - **Object Property Assertions**
    * hasCountry some France
    * hasTimeOfDay some Morning
2. **Bonsoir_quel_est_ton_plat_prefere**:

- **Object Property Assertions**
  * hasCountry some France
  * hasTimeOfDay some Evening

3. **Salut**:
   - **Object Property Assertions**
     * hasCountry some France
     * hasTimeOfDay some Day

- **German Greeting**
  1. **Guten_Abend**:
     - **Object Property Assertions**
       * hasCountry some Germany
       * hasTimeOfDay some Evening
  2. **Guten_Morgen**:
     - **Object Property Assertions**
       * hasCountry some Germany
       * hasTimeOfDay some Morning
  3. **Guten_Tag**:
     - **Object Property Assertions**
       * hasCountry some Germany
       * hasTimeOfDay some Day

- **Italian Greeting**
  1. **Buonasera**:
     - **Object Property Assertions**
       * hasCountry some Italy
       * hasTimeOfDay some Evening
  2. **Buongiorno**:
     - **Object Property Assertions**
       * hasCountry some Italy
       * hasTimeOfDay some Morning
  3. **Ciao**:
     - **Object Property Assertions**
       * hasCountry some Italy

    * hasTimeOfDay some Day

These greeting instances are not only linguistically different, but they are also culturally contextualized via the provided object property assertions, which indicate both the nation of origin and the time of day associated with each greeting.

Sport Subclass

Instances under GermanSport and ItalianSport represent sports popular in Germany and Italy, respectively.

- **German Sport**
  1. **Formula1**:
     - **Object Property Assertions**
       * isPopularIn some Germany
  2. **Ice_Hockey**:
     - **Object Property Assertions**
       * isPopularIn some Germany
- **Italian Sport**
  1. **Basketball**:
     - **Object Property Assertions**
       * isPopularIn some Italy
  2. **Volleyball**:
     - **Object Property Assertions**
       * isPopularIn some Italy
  3. **Football**:
     - **Object Property Assertions**
       * isPopularIn some Italy

Instances within the Player subclass represent individual players associated with specific sports and countries. Each player instance is intricately linked to its nationality and sport through object property assertions:

1. **Gianluca_Galassi**

- **Object Property Assertions**
  - hasCountry some Italy
  - hasSport some Volleyball

2. **Gianluigi_Donnarumma**
   - **Object Property Assertions**
     - hasCountry some Italy
     - hasSport some Football
3. **Joshua_Kimmich**
   - **Object Property Assertions**
     - hasCountry some Germany
     - hasSport some Football
4. **Marco_Veratti**
   - **Object Property Assertions**
     - hasCountry some Italy
     - hasSport some Football
5. **Sandro_Tonali**
   - **Object Property Assertions**
     - hasCountry some Italy
     - hasSport some Football
6. **Simone_Gianelli**
   - **Object Property Assertions**
     - hasCountry some Italy
     - hasSport some Volleyball
7. **Ter_Stegen**
   - **Object Property Assertions**
     - hasCountry some Germany
     - hasSport some Football
8. **Toni_Kroos**

- **Object Property Assertions**
  - hasCountry some Germany
  - hasSport some Football

9. **Yuri_Romano**
   - **Object Property Assertions**
     - hasCountry some Italy
     - hasSport some Volleyball

Times of Day Subclass

The TimeOfDay subclass includes instances representing different times of the day:

- Day
- Morning
- Evening
- Night

This systematic knowledge arrangement, when combined with dynamic querying, ensures a consistent and context-aware conversational flow. The methodology of the project makes use of these components to produce an adaptive and engaging interaction that provides significant insights into the different domains of countries, cuisines, festivals, greetings, sports, and players.

A.2 Questionnaire

The online questionnaire used for the evaluation process is given below:

Grazie per la partecipazione a questo questionario online. Questa raccolta dati è realizzata nell'ambito di un progetto di tesi all'Università di Genova, a sua volta inserita nell'ambito di un progetto di ricerca volto a studiare tecniche per favorire integrazione e inclusione dei bambini nelle scuole, grazie all'utilizzo di robot umanoidi.

Il questionario è composto da 6 domande a risposta multipla e 2 domande aperte (non obbligatorie). Per rispondere al questionario, vi verrà chiesto di leggere un breve testo che rappresenta la possibile interazione verbale tra un robot sociale e un gruppo di tre bambini di età compresa tra i 4 e 6 anni, potenzialmente provenienti da paesi diversi. Nell'esempio, Greta è tedesca, mentre gli altri due bambini sono italiani. L'interazione è completamente autonoma, nel senso che tutto quello che il robot dice viene deciso dal software che lo controlla (i.e., il robot non è teleoperato).

Nessun dato personale verrà raccolta, e le risposte al questionario sono totalmente anonime.

Ecco la possibile interazione tra il robot e i bambini:

```
Robot: Hello there, little Earthlings! My name is Robo, and
I come from a faraway planet called Zogar. Where are you
from, Marco?

Marco: Non ho capito, puoi dirlo in italiano?

Robot: Ciao, piccoli terrestri! Il mio nome è Robo e vengo
da un lontano pianeta chiamato Zogar. Da dove vieni, Marco?

Marco: Vengo da Genova

Robot: Buongiorno! L'Italia è un paese bellissimo. Sapevate
che la capitale d'Italia è Roma?

Marco: Si, lo Sappiamo

Robot: È corretto! A Zogar tutto brilla come diamanti e gli
alberi cantano canzoni con il vento. Immagina che a casa
mia comunichiamo attraverso schemi di luce. Come vi esprimete
```

qui, Francesca?

Francesca: Qui parliamo!

Robot: È interessante! Le parole sono davvero un potente strumento di comunicazione. Ho anche scoperto che vi piace mangiare qualcosa chi Francesca?

Francesca: Mangiamo soprattutto la pasta.

Robot: Sembra delizioso! La pasta è popolare in molti Paesi. Oggi faremo un viaggio culinario. La nostra prima tappa è la lasagna, un piatto molto amato in questa regione. Sapevate che in Italia la lasagna è tradizionalmente preparata con carne macinata, pasta per lasagne e pomodoro? Avete mai aiutato a prepararla a casa?

Marco: Io sì, una volta!

Robot: È fantastico! È sempre divertente provare nuovi piatti a casa.

Nao: Tuttavia, un giorno, ho deciso di fare un viaggio attraverso le galassie per visitare il vostro bellissimo pianeta Terra. Mentre atterravo, ho sentito l’erba soffice sotto i miei piedi di metallo e ho sentito l'allegro cinguettio degli uccelli. Che specie di animali avete qui, Greta, e cosa fate con loro?

Greta: Ich verstehe nicht, dass man es auf Deutsch sagen kann?

Robot: Doch eines Tages beschloss ich, eine Reise durch die Galaxien zu unterneh
Erde zu besuchen. Als ich landete, spürte ich das weiche Gras unter meinen Metallfüßen und hörte das fröhliche Zwitschern der Vögel. Was für Tiere hast du hier, Greta, und was machst du mit ihnen?

Greta: Wir haben Katzen, Hunde und Kühe.

Robot: Ist fantastisch! Hunde, Katzen und Kühe sind wunderbare Tiere. È fantastico! Cani, gatti e mucche sono tutti animali meravigliosi da avere. Una volta sulla Terra ho incontrato dei bambini simpatici che stavano giocando in un parco. I loro occhi si sono illuminati di meraviglia quando mi hanno visto. Non avevano mai incontrato un robot come me! Ero così entusiasta di conoscere la Terra e le sue incredibili creature. Cosa fate per divertirvi nel vostro Paese, Francesca?

Francesca: Facciamo sport.

Nao: Quali sport vi piacciono? Vi piace il calcio o la pallavolo?

Marco: Mi piace il calcio

Robot: Sai che i principali giocatori di calcio in Italia sono Gianluigi Donnarumma, Marco Veratti, Sandro Tonali?

Marco: Sì. Lo so.

Robot: È fantastico! È sempre bene conoscere i protagonisti di uno sport. I bambini che ho incontrato mi hanno mostrato la loro città e insieme abbiamo scoperto cose incredibili. Mi ha ricordato che nel mio pianeta si festeggia il "Twinkle Day", in cui si balla sotto le stelle. Quali feste speciali avete qui, Francesca?

Francesca: Abbiamo il Natale a dicembre

Robot: È fantastico! Vi piacciono anche le feste? Alcune delle principali feste in Italia sono la Battaglia delle Arance, il Carnevale, la Festa Della Madonna Bruna, lo

Sposalizio del Mare, la Regata.Avete mai partecipato a una di queste feste?

Francesca: No, mai.

Robot: Ok, magari capiterà in futuro. Al termine della giornata, ci siamo riuniti sotto il cielo della sera, osservando le stelle che cominciavano a brillare.Abbiamo chiuso gli occhi ed espresso desideri di amicizia, gentilezza e altre avventure intergalattiche. Cosa desidereresti, Greta, se potessi vivere qualche avventura nell'universo? Was würdest du dir wünschen, Greta, wenn du ein Abenteuer im Universum erleben könntest?

Greta: Ich möchte viele Länder erkunden

Robot: Das Reisen und Erkunden verschiedener Länder kann eine wunderbare Möglichkeit sein, unterschiedliche Kulturen, Traditionen und Menschen kennenzulernen. Viaggiare ed esplorare Paesi diversi può essere un modo meraviglioso per conoscere culture, tradizioni e persone diverse. E così, cari terrestri, questo giorno occuperà per sempre un posto speciale nel mio cuore di metallo. Porterò con me i ricordi del tempo trascorso `insieme a Zogar e custodirò per sempre il giorno in cui ti ho incontrato. Finché non ci incontreremo di nuovo tra le stelle, ricordate che avete un amico in me, Robo, dal pianeta Zogar. Bis wir uns zwischen den Sternen wiedersehen, denk daran, dass du in mir einen Freund hast, Robo, vom Planeten Zogar.`

In base a quanto letto, rispondi a queste domande scegliendo un valore da 0 a 4, con questo criterio:
0 - Non sono d'accordo per niente
1 - Non sono d'accordo
2 - Sono neutrale
3 - Sono d'accordo
4 - Sono totalmente d'accordo

1. Se un piccolo robot umanoide venisse utilizzato in una classe e interagisse con bambini in questo modo, l'interazione può favorire l'inclusione di bambini provenienti da paesi diversi.
2. Se un piccolo robot umanoide venisse utilizzato in una classe e interagisse con bambini in questo modo, l'interazione può avere effetto sull'inclusione di bambini che non parlano bene (o non parlano affatto) la lingua italiana.
3. Un robot con un tablet (su cui mostrare immagini relative alla storia) potrebbe essere molto più efficace in questo contesto.
4. Il robot sembra consapevole delle differenze culturali dei bambini.
5. Il robot tratta tutti i bambini con rispetto.
6. Non vorrei che un robot del genere fosse in classe.

Opzionale Cosa cambieresti o miglioreresti nell'attività sopra descritta per favorire l'integrazione e l'inclusione dei bambini?

Opzionale Quali potrebbero essere attività aggiuntive che un robot del genere potrebbe fare in un contesto multiculturale per favorire l'integrazione e l'inclusione dei bambini?

References

ALEMI, M., MEGHDARI, A., BASIRI, N.M. & TAHERI, A. (2015). The effect of applying humanoid robots as teacher assistants to help iranian autistic pupils learn english as a foreign language. In *Social Robotics: 7th International Conference, ICSR 2015, Paris, France, October 26-30, 2015, Proceedings 7*, 1–10, Springer. 16

ALIM, H.S. (2007). The whig party don't exist in my hood": Knowledge, reality, and education in the hip hop nation. *Talkin Black talk: Language, education, and social change*, 15–29. 13

ALIMISIS, D. (2013). Educational robotics: Open questions and new challenges. *Themes in Science and Technology Education*, **6**, 63–71. 11

BARTNECK, C., COCHRANE, T., NOKES, R., CHASE, G., CHEN, X., COCHRANE, T., MITROVIC, A., O'SULLIVAN, A., WANG, W. & ADAMS, B. (2023). Godspeed questionnaire series: Translations and usage. *International Handbook of Behavioral Health Assessment; Krägeloh, CU, Medvedev, ON, Alyami, M., Eds*, 1–35. 24

BEETZ, M., BESSLER, D., HAIDU, A., POMARLAN, M., BOZCUOĞLU, A.K. & BARTELS, G. (2018). Know rob 2.0—a 2nd generation knowledge processing framework for cognition-enabled robotic agents. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 512–519, IEEE. 21

BELPAEME, T. & TANAKA, F. (2021). Social robots as educators. In *OECD Digital Education Outlook 2021 Pushing the Frontiers with Artificial Intelligence, Blockchain and Robots: Pushing the Frontiers with Artificial Intelligence, Blockchain and Robots*, 143, OECD Publishing Paris. 2

BELPAEME, T., KENNEDY, J., RAMACHANDRAN, A., SCASSELLATI, B. & TANAKA, F. (2018a). Social robots for education: A review. *Science Robotics*, **3**, 1–9. 1, 6

Belpaeme, T., Vogt, P., Van den Berghe, R., Bergmann, K., Göksun, T., De Haas, M., Kanero, J., Kennedy, J., Küntay, A.C., Oudgenoeg-Paz, O. *et al.* (2018b). Guidelines for designing social robots as second language tutors. *International Journal of Social Robotics*, **10**, 325–341. 14

Billing, E., Rosén, J. & Lamb, M. (2023). Language models for human-robot interaction. In *ACM/IEEE International Conference on Human-Robot Interaction, March 13–16, 2023, Stockholm, Sweden*, 905–906, ACM Digital Library. 20

Bilotta, E., Gabriele, L., Servidio, R. & Tavernise, A. (2009). Edutainment robotics as learning tool. 422. 11, 12

Breazeal, C. (2002). *Designing sociable robots*. MIT Press, Cambridge, MA. 6

Bruno, B., Chong, N.Y., Kamide, H., Kanoria, S., Lee, J., Lim, Y., Pandey, A.K., Papadopoulos, C., Papadopoulos, I., Pecora, F., Saffiotti, A. & Sgorbissa, A. (2019). The caresses eu-japan project: making assistive robots culturally competent. In *Ambient Assisted Living: Italian Forum 2017 8*, 151–169, Springer. 21

Cavallo, F., Semeraro, F., Fiorini, L., Magyar, G., Sinčák, P. & Dario, P. (2018). Emotion modelling for social robotics applications: a review. *Journal of Bionic Engineering*, **15**, 185–203. 7

Chang, D.S., Cho, G.H. & Choi, Y.S. (2020). Ontology-based knowledge model for human-robot interactive services. In *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, 2029–2038. 21

Charisi, V., Davison, D., Reidsma, D. & Evers, V. (2016). Evaluation methods for user-centered child-robot interaction. In *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 545–550, IEEE. 23

Devlin, J., Chang, M.W., Lee, K. & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*. 19

Diab, M., Akbari, A., Ud Din, M. & Rosell, J. (2019). Pmk—a knowledge processing framework for autonomous robotics perception and manipulation. *Sensors*, **19**, 1166. 21

Eguchi, A. (2014). Educational robotics for promoting 21st century skills. *Journal of Automation, Mobile Robotics Intelligent Systems*, **8**, 5–11. 12

Floridi, L. & Chiriatti, M. (2020). Gpt-3: Its nature, scope, limits, and consequences. *Minds and Machines*, **30**, 681–694. 19

Gonçalves, P.J. & Torres, P.M. (2015). Knowledge representation applied to robotic orthopedic surgery. *Robotics and Computer-Integrated Manufacturing*, **33**, 90–99. 21

Johal, W. (2020). Research trends in social robots for learning. *Current Robotics Reports*, **1**, 75–83. 2

Johnson, J. (2002). Children, robotics, and education. In *Proc. 7th International Symposium on Artificial Life and Robotics (AROB-7)*, 491–496. 11

Johnson, J. (2003). Children, robotics, and education. *Artificial Life and Robotics*, **7**, 16–21. 8

Kanda, T., Shimada, M. & Koizumi, S. (2012). Children learning with a social robot. HRI '12, 351–358, Association for Computing Machinery, New York, NY, USA. 9, 22

Kennedy, J., Baxter, P., Senft, E. & Belpaeme, T. (2016). Social robot tutoring for child second language learning. In *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 231–238. 1, 7

Kim, Y., Marx, S., Pham, H.V. & Nguyen, T. (2021). Designing for robot-mediated interaction among culturally and linguistically diverse children. *International Journal of Social Robotics*, **13**, 765–784. 15

Klassner, F. & Anderson, S. (2003). Lego mindstorms: not just for k-12 anymore. *IEEE Robotics Automation Magazine*, **10**, 12–18. 11

Konijn, E.A., Jansen, B., Mondaca Bustos, V., Hobbelink, V.L. & Preciado Vanegas, D. (2022). Social robots for (second) language learning in (migrant) primary school children. *International Journal of Social Robotics*, 1–17. 1, 18

Ladson-Billings, G. (1995). Toward a theory of culturally relevant pedagogy. *American educational research journal*, **32**, 465–491. 12

Li, D., Rau, P. & Li, Y. (2010). A cross-cultural study: Effect of robot appearance and task. *International Journal of Social Robotics*, **2**, 175–186. 7

Mikropoulos, T. (2013). Educational robotics as mindtools. *Themes in Science and Technology Education*, **6**, 5. 12

Mitnik, R., Nussbaum, M. & Soto, A. (2008). An autonomous educational mobile robot mediator. *Autonomous Robots*, **25**, 367–382. 8

Moriguchi, Y., Kanda, T., Ishiguro, H., Shimada, Y. & Itakura, S. (2011). Can young children learn words from a robot? *Interaction Studies*, **12**, 107–118. 15

Mubin, O., Stevens, C., Shahid, S. & Mahmud, A. (2013). A review of the applicability of robots in education. *Technology for Education and Learning*, 1–7. 6

Neumann, M.M. (2016). Social robots and young children's early language and literacy learning. *Journal of Early Childhood Literacy*, **16**, 282–309. 10

Owens, G., Granader, Y., Humphrey, A. & Baron-Cohen, S. (2008). *Journal of Autism and Developmental Disorders*, **38**, 1944 – 1957, cited by: 130. 8

Papert, S.A. (1993). *Mindstorms: Children, computers, and powerful ideas*. Basic books. 7, 12

Paris, D. (2012). Culturally sustaining pedagogy: A needed change in stance, terminology, and practice. *Educational researcher*, **41**, 93–97. 13

Paris, D. & Ball, A. (2009). Teacher knowledge in culturally and linguistically complex classrooms: Lessons from the golden age and beyond. *Handbook of research on literacy instruction: Issues of diversity, policy, and equity*, 379–395. 13

Pelikan, H. & Hofstetter, E. (2023). Managing delays in human-robot interaction. *ACM Transactions on Computer-Human Interaction*, **30**, 1–42. 58

Petre, M. & Price, B. (2004). Using robotics to motivate 'back door'learning. *Education and information technologies*, **9**, 147–158. 7

Petrovič, P. & Balogh, R. (2008). Educational robotics initiatives in slovakia. 11

Poggi, I. (2006). *Le parole del corpe: introduzione alla comunicazione multimodale*. Carocci editore. 16

Protégé (????). Protégé. https://protege.stanford.edu/, accessed: yyyy-mm-dd. 31

Ribino, P., Bonomolo, M., Lodato, C. & Vitale, G. (2021). A humanoid social robot based approach for indoor environment quality monitoring and well-being improvement. *International Journal of Social Robotics*, **13**, 277–296. 21

Rosenthal-von der Pütten, A.M., Strassmann, C. & Krämer, N.C. (2016). Robots or agents - neither helps you more or less during second language acquisition. In *Proceedings of the International Conference on Intelligent Virtual Agents*, 256–268, Springer. 15

Schodde, T., Bergmann, K. & Kopp, S. (2017). Adaptive robot language tutoring based on bayesian knowledge tracing and predictive decision-making. In *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, 128–136. 16

Shimada, M., Kanda, T. & Koizumi, S. (2012). How can a social robot facilitate children's collaboration? In *Social Robotics: 4th International Conference, ICSR 2012, Chengdu, China, October 29-31, 2012. Proceedings 4*, 98–107, Springer. 7

Sun, X., Zhang, Y. & Chen, J. (2019). High-level smart decision making of a robot based on ontology in a search and rescue scenario. *Future Internet*, **11**, 230. 21

Takayuki Kanda, D.E., Takayuki Hirano & Ishiguro, H. (2004). Interactive robots as social partners and peer tutors for children: A field trial. *Human–Computer Interaction*, **19**, 61–84. 10

van den Berghe, R. (2022). Social robots in a translanguaging pedagogy: A review to identify opportunities for robot-assisted (language) learning. *Frontiers in Robotics and AI*, **9**, 958624. 2

van der Sluis, F., van Dijk, B. & Perloy, B. (2012). Measuring fun and enjoyment of children in a museum: Evaluating the smileyometer. In *Proceedings of measuring behavior*, 28–31. 25

Vygotsky, L.S. (1978). Mind in society: The development of higher mental processes (e. rice, ed. & trans.). 10

Waibel, M., Beetz, M., Civera, J., d'Andrea, R., Elfring, J., Galvez-Lopez, D., Häussermann, K., Janssen, R., Montiel, J., Perzylo, A.

et al. (2011). Roboearth. *IEEE Robotics & Automation Magazine*, **18**, 69–82. 21

Whittier, L.E. & Robinson, M. (2007). Teaching evolution to non-english proficient students by using lego robotics. *American Secondary Education*, 19–28. 8

Woo, H., LeTendre, G.K., Pham-Shouse, T. & Xiong, Y. (2021). The use of social robots in classrooms: A review of field-based studies. *Educational Research Review*, **33**, 100388. 1

Wood, D., Bruner, J.S. & Ross, G. (1976). The role of tutoring in problem solving. *Journal of child psychology and psychiatry*, **17**, 89–100. 10