Application of Reinforcement Learning and Transformers for Language Learning

Master's Thesis,
March 2025

Master in Artificial Intelligence

Second Edition,
Academic Year 2024

By

Julio Emanuel Suriano Bryk
ID 37.620.411

Supervised by

José Gabriel García Pardo

Education is not about thinning the herd.
Education is about helping every student succeed.

Andrew Ng

Acknowledgments

I would like to thank the professors and colleagues of the master's program for their support, valuable contributions, and exchanges of ideas that have enriched this work.

I also want to highlight my partner for their understanding and support throughout the process, which allowed me to stay focused and motivated.

Finally, I thank my family and friends for their constant support and for always being available to offer their help and perspective.

Contents

List of Figures

List of Tables

List of Algorithms

Glossary

AI Artificial Intelligence - Set of technologies that allow machines to learn, reason, and make decisions.

Assistant UI Open-source framework for creating conversational chat interfaces.

Attention Mechanism Key component of the Transformer architecture that allows the model to focus on different parts of the input according to their relevance.

Beam Search Heuristic search algorithm that explores a graph by building the graph gradually from the root, expanding the most promising node in a limited set of nodes.

CEFR Common European Framework of Reference for Languages.

Data Mining Process of discovering patterns and relationships in large data sets.

Feed-Forward Neural network layer that applies a linear transformation followed by an activation function.

Generator Component of an information retrieval system that creates responses from retrieved documents.

Hallucinations Errors in text generation that result in incoherent or incorrect responses.

ITS Intelligent Tutoring System.

Knowledge Base Structured data set that stores relevant information for an information retrieval system.

Likert Scale Psychometric measurement method that evaluates attitudes and opinions through a response scale with a range of options.

LLM Large Language Model - Large-scale language model.

Machine Learning Branch of artificial intelligence that allows systems to learn and improve from experience.

MDP Markov Decision Process - Mathematical framework for modeling decision-making in situations where outcomes are partly random and partly under control.

MFCC Mel-Frequency Cepstral Coefficients. Coefficients that represent the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a nonlinear mel scale of frequency.

Multi-Agent System System composed of multiple intelligent agents that interact with each other to solve complex problems.

NLP Natural Language Processing.

Open Source Software whose source code is publicly available and can be modified and distributed by anyone.

Policy Strategy that an agent follows to determine its actions based on the current state of the student.

PPO Proximal Policy Optimization - Reinforcement learning algorithm that optimizes control policies in continuous and stochastic decision environments.

PyTorch Open-source deep learning library developed by Meta.

RAG Retrieval-Augmented Generation - System that combines information retrieval with text generation.

Recommendation System System that suggests relevant content based on the user's profile and behavior.

REST API Application programming interface based on the HTTP protocol and the GET, POST, PUT, and DELETE request methods.

Retriever Component of an information retrieval system that selects relevant documents based on a query.

Reward Function Function that defines the feedback an agent receives based on student progress.

RL Reinforcement Learning - A branch of AI that allows systems to learn through interaction with an environment.

Self-Attention Mechanism that allows a model to evaluate relationships between all positions in a sequence.

SLA Second Language Acquisition - Process of acquiring a second language.

STT Speech-to-Text. System that converts speech into written text through automatic speech recognition.

TensorFlow Open-source library for machine learning developed by Google.

Transformers Neural network architecture that has revolutionized natural language processing.

TTS Text-to-Speech. System that converts written text into synthesized speech.

Viterbi Algorithm that finds the most likely sequence of hidden states in a hidden Markov model, commonly used in speech recognition for decoding.

Abstract

This Master's Thesis presents an innovative language learning system that integrates Reinforcement Learning (RL), Transformers architectures, and Retrieval-Augmented Generation (RAG) technologies to optimize the educational experience. The system implements a Proximal Policy Optimization (PPO) algorithm that dynamically adapts content according to the student's profile, generating personalized learning paths with accuracy exceeding 95

The integration of Large Language Model (LLM) models with voice processing (Text-to-Speech (TTS) and Speech-to-Text (STT)) enables the creation of interactive dialogues and realistic conversational simulations, where students can simultaneously develop listening comprehension and oral production skills. The system's architecture is designed to be scalable and flexible, allowing the integration of new modules and functionalities.

Introduction 1

Language learning in the digital era has undergone a significant transformation thanks to advances in the field of Artificial Intelligence (AI). However, one of the greatest challenges remains the effective personalization of the learning process to adapt to the individual needs of each student. This work proposes an innovative approach that combines RL techniques with Transformers architectures and voice processing technologies to create an adaptive and personalized language learning system.

1.1 Document Structure

This thesis is organized into seven chapters that guide the reader from the theoretical foundations to the final results and conclusions:

Chapter 1: Introduction Presents the motivation, current limitations, opportunities for improvement, and objectives of the work.

Chapter 2: State of the Art Reviews the most advanced technologies and systems in the field of AI-assisted language learning.

Chapter 3: Theoretical Framework Explores the theoretical foundations of language learning, artificial intelligence in education, natural language processing, and reinforcement learning.

Chapter 4: Materials Details the technological resources, infrastructure, and tools used in the development of the system.

Chapter 5: Methods Describes the system architecture, component implementation, and the reinforcement learning model for level adaptation.

Chapter 6: Results Presents the results obtained, system evaluation, and analysis of preliminary tests.

Chapter 7: Conclusions Analyzes the achievements, contributions made, limitations identified, and future lines of research.

Additionally, two technical appendices are included that delve into specific aspects of the voice processing technologies used: Faster Whisper for speech recognition (Appendix A) and Kokoro TTS for speech synthesis (Appendix B).

1.2 Motivation

Second language acquisition is a complex process that varies significantly among individuals. This process is influenced by multiple factors, such as learning style, previous experiences, level of motivation, and specific aptitudes of each student Ellis (1994). Traditional language teaching methods, even in their digitized form, present significant limitations that prevent effective personalization and dynamic adaptation to student progress.

Current systems often follow a predefined sequential model that does not adequately consider individual differences, which can result in inefficient or demotivating learning experiences. As Krashen (1982) points out, optimal learning occurs when the input is slightly above the student's current level (i+1 principle), a balance that is difficult to achieve with systems that do not adapt dynamically.

1.2.1 Current Limitations

Currently, language teaching methods face several limitations that affect learning effectiveness. These limitations can be classified into four main categories:

- **Structural Rigidity**: Programs follow predefined sequences that do not adapt to the student's actual progress, limiting the ability to respond to their specific needs.
- **Lack of Personalization**: They do not adequately consider different learning styles, interests, and individual preferences, which can affect motivation and learning effectiveness.
- **Limited Feedback**: Most systems provide basic feedback without considering the complete context of learning, making it difficult to identify specific areas for improvement.
- **Artificial Conversational Practice**: Interactions are often mechanical and do not reflect the dynamic nature of real language, limiting the student's ability to apply their skills in real-life situations.

These limitations highlight the need for a more flexible and personalized approach to language teaching, which can adapt to the individual needs and progress of each student, providing a more effective and motivating learning experience.

1.2.2 Opportunities for Improvement

Recent advances in artificial intelligence, particularly in the field of natural language processing and reinforcement learning, open new possibilities to overcome the previously mentioned limitations. Four main areas of opportunity are identified:

- **Dynamic Adaptability:** Implement systems that adjust content and difficulty in real-time, based on the student's performance and needs. RL algorithms, as demonstrated

by Williams y Chen (2017), are particularly suited for this task, as they can optimize sequential decisions in learning environments.

- **Deep Personalization:** Consider multiple individual factors, such as learning style, interests, and pace of progress, to optimize the learning process. Modern architectures based on Transformers allow analyzing complex behavior patterns and adapting the educational experience in a more granular way Vaswani et al. (2017).
- **Natural Interaction:** Use advanced technologies, such as natural language models and voice processing, to simulate more realistic and dynamic conversations. Recent advances in LLM Brown et al. (2020) and voice technologies Graves et al. (2013) allow for much more natural interactions than previous systems.
- **Contextual Feedback:** Provide detailed and specific feedback, based on the context and profile of the student, to improve understanding and performance. RAG systems Lewis et al. (2020) can significantly enrich the quality and relevance of this feedback.

The combination of these advanced technologies offers transformative potential for the field of language learning, allowing the creation of adaptive systems that dynamically respond to the individual needs of each student.

1.3 Objectives

Based on the motivation presented and the opportunities identified, this work establishes the following objectives:

1.3.1 General Objective

Develop a language learning system that integrates RL, Transformers architectures, and a Multi-Agent System approach to provide a personalized, adaptive, and effective learning experience that overcomes the limitations of traditional methods and leverages the capabilities of the most recent artificial intelligence technologies.

1.3.2 Specific Objectives

To achieve the general objective, several specific objectives have been defined that focus on the implementation of advanced AI techniques. These specific objectives are organized into four main areas:

1.3.2.1 Learning Optimization

- Implement a PPO algorithm that optimizes personalized learning paths according to the student's profile and progress.
- Develop mechanisms for dynamic content adaptation that adjust difficulty in real-time.

- Create continuous evaluation systems that measure progress in multiple linguistic dimensions.

1.3.2.2 Interaction Enhancement

- Integrate advanced LLM for Natural Language Processing (NLP) that allow deep contextual understanding.
- Develop dialogue systems that reproduce natural and contextually relevant conversations.
- Implement real-time error analysis with specific and constructive feedback.

1.3.2.3 Language Skills Improvement

- Create pronunciation evaluation systems using advanced TTS and STT technologies.
- Develop adaptive comprehension exercises that evolve according to the student's level.
- Implement contextualized conversational practice that simulates real language use situations.

1.3.2.4 Knowledge Management

- Integrate RAG systems for efficient and contextualized access to relevant educational resources.
- Develop dynamic knowledge bases that evolve with the student's needs.
- Implement automatic content update mechanisms to keep resources up-to-date.

State of the Art

2

This section presents a detailed review of the most advanced technologies and systems in the field of Artificial Intelligence (AI)-assisted language learning. The objective is to contextualize the present research within the current landscape, identifying trends, significant advances, and opportunities for innovation. Six key technological areas are analyzed: adaptive learning systems, LLM applications, AI-based learning companions, advances in voice processing, multi-agent systems, and RL frameworks. Finally, this chapter explains how these technologies are integrated into the proposal of this work.

2.1 Adaptive Learning Systems

Modern language learning systems have evolved significantly in recent years, incorporating advanced AI and machine learning algorithms that allow precise adaptation to each user's level and needs. This evolution represents a paradigm shift compared to traditional static approaches, enabling personalized and dynamic educational experiences Roll y Wylie (2018).

Below, the most innovative platforms in the market are analyzed, highlighting their main technological features and pedagogical approaches:

- **Busuu Conversations (2024)**[1]: Incorporates an AI system that analyzes error patterns and dynamically adjusts content to improve learning effectiveness.
- **Duolingo Max (2024)**[2]: Uses GPT-4 to generate personalized explanations and maintain contextual conversations, adapting to the user's level.
- **Babbel Everyday Conversations (2023)**[3]: Combines AI with human tutors to optimize the hybrid learning experience, offering more personalized interaction.
- **Lingvist (2023)**[4]: Uses contextual data to generate exercises, lessons, and adapted recommendations, facilitating the retrieval of relevant linguistic content and the generation of interactive activities.

[1] https://www.busuu.com
[2] https://www.duolingo.com
[3] https://www.babbel.com
[4] https://www.lingvist.com

- **Elsa Speak (2023)**[5]: AI-assisted pronunciation system that provides real-time feedback and personalized exercises to improve fluency and pronunciation accuracy.

These systems represent the current state of the art in adaptive language learning, but as VanLehn (2011) points out, there are still significant challenges regarding precise modeling of student knowledge and adaptation to diverse learning styles, areas where this research seeks to contribute.

2.2 LLM Applications in Language Education

LLMs have radically transformed the landscape of language learning, providing unprecedented capabilities for natural dialogue generation, contextual analysis, and intelligent correction. This section analyzes the main applications of these technologies, categorized into conversational systems and textual analysis tools.

2.2.1 Assistants and Dialogue

The evolution of conversational systems has reached a level of sophistication that allows almost human-like interactions, offering highly effective language practice environments:

- **ChatGPT (2022)**[6]: Revolutionized human-AI interaction by establishing the standard for natural conversational interfaces and creating a complete development ecosystem.
- **Claude (2023)**[7]: Stood out for its superior accuracy in document analysis and ability to follow complex instructions with less tendency to hallucinate.
- **Azure Language Studio (2023)**[8]: Offers linguistic analysis tools and educational content generation, improving the quality of learning.
- **LLaMA (2023)**[9]: Open Source model developed by Meta, designed to be efficient and accessible for research and practical applications.

2.2.2 Analysis and Correction

Analysis and correction tools based on LLMs have evolved beyond simple identification of grammatical errors, incorporating deep contextual understanding and stylistic recommendations:

- **Grammarly with GrammarlyGO (2023)**[10]: Uses generative AI to provide contextual corrections and improvement suggestions, helping users write with greater precision.

[5] https://www.elsaspeak.com
[6] https://chatgpt.com/
[7] https://claude.ai/
[8] https://language.cognitive.azure.com/
[9] https://ai.facebook.com/blog/large-language-model-llama
[10] https://www.grammarly.com

- **DeepL Write (2023)**[11]: Correction system that considers cultural context and linguistic register, offering more relevant and precise suggestions.

The advancement of these systems, however, poses important challenges related to excessive dependence on automated correction and the potential impact on learning autonomy Rodríguez et al. (2023), aspects that must be carefully considered in the development of new educational systems based on LLMs.

2.3 Emerging Technologies with Learning Companions

AI-based learning companions represent a significant evolution in educational systems, implementing a social and emotional dimension that complements the transmission of technical knowledge. These systems go beyond simple instruction, establishing a pedagogical relationship that includes motivation, personalized adaptation, and constant support (Baker y Inventado, 2014).

- **Khanmigo (2024)**[12]: Khan Academy's virtual tutor that acts as a personalized study companion, providing adaptive explanations, step-by-step guidance, and instant feedback across multiple subjects.
- **Third Space Learning (2024)**[13]: Platform that combines human tutors with AI to create a hybrid learning experience, where the system analyzes interactions and provides personalized insights.
- **Riiid SANTA (2023)**[14]: Adaptive tutoring system for predicting student performance and personalizing content, maximizing learning efficiency through predictive analysis.

These learning companion systems represent a promising direction for the future of language education, as they provide a personalized and adaptive practice environment that can significantly complement traditional methods.

2.4 Advances in Voice Processing

Voice processing technologies, including TTS and STT, have experienced revolutionary advances in recent years, radically transforming the possibilities for pronunciation learning and listening comprehension. These systems have evolved from robotic voices and limited recognition to achieve near-human levels of naturalness and precision (Graves et al., 2013).

- **Whisper OpenAI (2022)**[15]: High-precision multilingual voice recognition, effective in noisy environments and with diverse accents. It is Open Source and used for automatic transcription and voice analysis in multiple languages.

[11] https://www.deepl.com/write
[12] https://www.khanacademy.org/khan-labs
[13] https://thirdspacelearning.com
[14] https://riiid.com
[15] https://openai.com/research/whisper

- **Google Speech-to-Text/Text-to-Speech (2023)**[16]: Real-time voice recognition with high accuracy, support for multiple languages, and easy integration with other Google platforms. Commonly used in virtual assistants and live meeting transcription.
- **Microsoft Azure AI Speech (2023)**[17]: Precise and fast transcription, with advanced capabilities for personalization and context adaptation. Ideal for customer service systems and real-time conversation analysis.
- **Deepgram (2023)**[18]: Voice recognition platform based on deep neural networks, known for its speed and precision. Used for call transcription and business conversation analysis.
- **Kokoro-82M (2025)**[19]: Kokoro is an open-source TTS model with 82 million parameters. Despite its lightweight architecture, it offers quality comparable to larger models, being significantly faster and more cost-effective.

These advances in voice processing open new possibilities for creating immersive conversational practice environments, where students can develop communication skills in realistic contexts with instant and personalized feedback.

2.5 Agentic AI

Multi-Agent System technology is becoming a key area of innovation in language learning. These technologies allow the creation of autonomous agents that can interact with each other and with users to provide more dynamic and personalized learning experiences. The multi-agent approach overcomes the limitations of monolithic systems by distributing responsibilities among agents with specific roles, improving both the effectiveness and robustness of the system Liu et al. (2023).

- **LangChain (2022)**[20]: Open Source platform that facilitates the creation of Multi-Agent System systems. LangChain allows the integration of different language models and specialized agents for specific tasks, improving system interaction and adaptability.
- **CrewAI (2023)**[21]: Open Source multi-agent system designed for team collaboration, allowing users to work together on language learning projects and receive real-time feedback.
- **phiData (2023)**[22]: Open Source platform that uses specialized agents to analyze linguistic data and provide personalized recommendations to improve language learning.

[16] https://cloud.google.com/speech-to-text
[17] https://azure.microsoft.com/en-us/products/ai-services/ai-speech
[18] https://deepgram.com
[19] https://huggingface.co/hexgrad/Kokoro-82M
[20] https://www.langchain.com
[21] https://www.crewai.com
[22] https://www.phidata.com

- **Autogen by Microsoft (2023)**[23]: Microsoft's Open Source technology that enables the creation of autonomous agents for specific tasks in language learning, improving the personalization and effectiveness of the educational process.

The autonomous agent paradigm represents a promising direction for the development of next-generation educational systems, allowing the creation of adaptive ecosystems that simulate the complex pedagogical roles that have traditionally been exclusive to human instructors.

2.6 Reinforcement Learning Frameworks

RL has proven to be a particularly suitable paradigm for the development of adaptive educational systems, thanks to its intrinsic ability to optimize strategies through sequential interactions, similar to the natural process of human learning Williams y Chen (2017). Modern RL frameworks provide robust tools to implement these systems at scale.

- **TensorFlow Agents (2019)**[24]: A RL library based on TensorFlow that provides tools to build, train, and evaluate RL agents. It is compatible with a wide range of algorithms and environments.

- **Stable Baselines3 (2020)**[25]: An implementation of RL algorithms in PyTorch, designed to be easy to use and extend. It is widely used for experimentation and development of RL solutions.

- **TorchRL (2022)**[26]: A reinforcement learning framework based on PyTorch, designed to be flexible and easy to use. It provides tools to build, train, and evaluate RL agents in various environments.

The choice of Stable Baselines3 for the implementation of the system proposed in this work is based on its optimal balance between ease of use and flexibility, as well as its robust implementation of the PPO algorithm, which has proven to be particularly effective for educational sequence optimization problems (Schulman et al., 2017).

2.7 Application of Technologies in This Work

This work integrates the most advanced technologies identified in the state of the art to develop a comprehensive language learning system. The proposal synthesizes multiple technological approaches into a cohesive and synergistic architecture, where each component contributes specific capabilities to the overall system.

[23] https://www.microsoft.com/en-us/research/project/autogen
[24] https://www.tensorflow.org/agents
[25] https://stable-baselines3.readthedocs.io
[26] https://github.com/pytorch/rl

- **Adaptive Learning Systems**: A system is implemented that analyzes user error patterns and dynamically adjusts content. This approach is inspired by the adaptive capabilities of Busuu Conversations, but incorporates multidimensional knowledge modeling that considers interdependencies between different linguistic skills.
- **LLM Applications**: A LLM is used to generate dialogues and provide contextual corrections. The integration of Phi-4, specifically optimized for the educational context, combines natural conversational capabilities with precision in linguistic evaluation and feedback.
- **Emerging Technologies with Learning Companions**: A virtual assistant is developed that acts as a learning companion, providing personalized and adaptive support. Inspired by Khanmigo's architecture, it implements scaffolding strategies based on the student's current level and detected learning style.
- **Advances in Voice Processing**: TTS and STT technology is integrated to enhance user interaction with the system. The implementation combines Faster-Whisper for speech recognition and Kokoro-TTS for synthesis, providing a naturalistic and precise oral communication experience.
- **Agentic AI**: The creation of autonomous agents that interact with each other and with users is explored. The system implements a multi-agent architecture based on LangChain, where specialized agents collaborate in different aspects of the educational process: tutoring, evaluation, motivation, and conversational practice.
- **Reinforcement Learning Frameworks**: RL frameworks are used to optimize the learning process and adapt content to user needs. Specifically, the PPO algorithm is implemented through Stable Baselines3 to optimize learning paths and dynamic level adaptation.

These technologies and theories are integrated into a unified system that overcomes the limitations of fragmented approaches, providing a language learning experience that is adaptive, interactive, and highly personalized, significantly improving both educational effectiveness and user experience.

This State of the Art analysis provides the contextual basis for understanding how our proposal is situated in the current landscape of technologies for language learning. The next chapter will delve into the theoretical framework that underpins the various components of the system, establishing the pedagogical and computational principles that guide its design.

Theoretical Framework 3

3.1 Fundamentals of Language Learning

3.1.1 Language Acquisition Theories

The field of Second Language Acquisition (SLA) has evolved significantly in recent decades, moving from behaviorist approaches to more cognitive and sociocultural perspectives, and more recently, towards the integration of AI technologies and adaptive systems that promise to revolutionize the way languages are learned.

Among the most influential theories in second language acquisition, the work of Krashen (1982) stands out, who developed the Monitor Model. This model includes five fundamental hypotheses, the most relevant being the comprehensible input hypothesis, which establishes that acquisition occurs when students receive input slightly above their current level of competence.

For his part, Ellis (1994) proposes a more integrative theoretical framework, emphasizing the interaction between cognitive and environmental factors in language learning. His work highlights the importance of considering both internal mental processes and contextual variables that influence language acquisition, providing a solid theoretical foundation for understanding how students process and acquire a second language.

3.1.2 Factors Influencing Second Language Learning

Ellis (1994) identifies various factors that affect language learning, which can be classified as internal and external.

Internal factors include the learner's age, linguistic aptitude, motivation and attitude, cognitive styles and learning strategies, as well as personality traits. Age influences brain plasticity and natural language acquisition capacity, while linguistic aptitude varies among individuals and can predict success in learning. Motivation can be intrinsic or extrinsic, and cognitive styles and learning strategies determine how information is processed and retained. Personality traits, such as extroversion, affect the willingness to participate in communicative interactions.

On the other hand, external factors include social and cultural context, exposure to the target language, and the quality and quantity of input. The learning environment and sociocultural context significantly influence attitudes toward the target language and its speakers,

largely determining learning success.

Frequent and varied exposure to the language is fundamental for developing linguistic competence, and the input must be comprehensible yet challenging, following the i+1 principle of Krashen (1982). This principle suggests that optimal learning occurs when the student is exposed to content slightly above their current level of competence.

Additionally, factors such as socioeconomic status, access to educational and technological resources, and linguistic policies of the environment also significantly influence the learning process. The availability of authentic materials and modern technological tools can considerably enrich the learning experience and facilitate exposure to the target language in meaningful contexts.

3.1.3 Teaching Methodologies

The evolution of teaching methodologies reflects our changing understanding of the language learning process:

3.1.4 Traditional Methods

The Grammar-Translation Method, predominant during the 19th and early 20th centuries Richards y Rodgers (2000), focuses on detailed analysis of grammatical rules and the translation of texts. This method emphasizes grammatical accuracy and reading comprehension, although it has been criticized for its limited attention to oral communication skills.

The Direct Method, introduced by Gouin (1892), emerged as a response to the limitations of the previous method, promoting total immersion in the target language and avoiding the use of the native language. This approach emphasizes the importance of oral communication and the direct association between language and meaning, without resorting to translation.

The Audiolingual Method, developed during World War II and founded by Fries (1945), is based on behaviorist principles and emphasizes the formation of linguistic habits through repetition and reinforcement. This method uses pattern drills and memorized dialogues to develop automaticity in language use.

3.1.5 Modern Approaches

The Communicative Language Teaching Approach Hymes (1972) marked a revolutionary change in language teaching by emphasizing communicative competence over mere grammatical accuracy. This approach fundamentally transformed the way languages are taught, prioritizing meaningful interactions and language use in real contexts.

Task-Based Learning Nunan (1989) represents another fundamental pillar, organizing learning around authentic communicative activities. Its effectiveness lies in promoting natural language learning while students focus on completing practical and meaningful tasks.

Content and Language Integrated Learning Coyle et al. (2010) has proven particularly effective by integrating academic content learning with language acquisition. This dual approach

not only improves learning efficiency but also significantly increases student motivation by providing a relevant context and clear purpose for language use.

3.1.6 Challenges in Learning Personalization

Learning personalization represents one of the greatest challenges in language teaching. As Ellis (1994) points out, a first fundamental challenge is the precise identification of the student's level, which requires comprehensive assessments that consider not only grammatical and lexical knowledge but also communicative skills in various contexts.

Adapting content to different learning styles constitutes another significant challenge, as it involves developing materials and activities that meet the individual preferences and needs of students, considering their different ways of processing and retaining linguistic information. Krashen (1982) emphasizes the importance of providing comprehensible input adapted to each student's individual level.

Maintaining motivation requires a delicate balance between challenge and support, needing strategies that maintain the student's interest and commitment over time. This is closely related to tracking individual progress, which must be continuous and detailed to allow timely adjustments in the learning process.

The scalability of personalized attention presents a particular challenge in educational contexts with limited resources, where it is necessary to find efficient ways to provide individualized feedback and personalized support to a large number of students simultaneously. This specific challenge motivates the implementation of AI-based systems, particularly those using RL and Transformers architectures, which can provide personalized attention at scale while maintaining instruction quality.

3.1.7 Progress Evaluation

Effective progress evaluation in language learning requires a multidimensional and systematic approach. Ellis (1994) emphasizes that communicative competence, which encompasses both linguistic knowledge and the ability to use it appropriately in social contexts, must be evaluated through tasks that reflect authentic communicative situations.

Grammatical accuracy, although it should not be the sole focus of evaluation, needs to be monitored to ensure that students develop an adequate mastery of fundamental linguistic structures. This evaluation must be balanced with the measurement of fluency, which reflects the student's ability to communicate effectively and naturally in real-time.

Krashen (1982) maintains that listening and reading comprehension require specific evaluations that consider different types of texts and discourses, as well as various communicative purposes. These evaluations must measure both global comprehension and the ability to identify specific details.

Progress evaluation and contextual feedback are crucial elements that can significantly benefit from the integration of advanced technologies. Systems based on LLM and TTS and STT technologies can provide more precise and detailed evaluations of the student's linguistic

skills. These systems can analyze error patterns, identify areas for improvement, and provide personalized feedback in real-time, overcoming the limitations of traditional evaluation methods.

3.2 Artificial Intelligence in Education

The integration of AI in the educational field has fundamentally transformed the way the teaching-learning process is conceived and implemented. This section explores the evolution and current state of intelligent educational systems, with special emphasis on their application in language teaching.

3.2.1 Evolution of Adaptive Learning Systems

Adaptive learning systems have evolved significantly since the first Intelligent Tutoring System (ITS) of the 1970s. VanLehn VanLehn (2011) notes that this evolution has gone through three main generations: rule-based systems, domain knowledge-based systems, and modern adaptive systems that use machine learning and AI techniques.

The first generation was characterized by systems that followed predefined rules to adapt content. The second generation incorporated more sophisticated domain models and began to consider the student's cognitive state. The current generation uses advanced AI techniques to create truly personalized learning experiences, capable of adapting in real-time to the student's needs and progress.

3.2.2 Architectures of Intelligent Educational Systems

Modern intelligent educational systems are built on modular architectures that integrate multiple specialized components. Anderson y Boyle (2020) identify four main components:

1. The expert module contains domain knowledge and pedagogical rules that guide instruction.
2. The student module maintains an updated model of the learner's knowledge and skills.
3. The pedagogical module determines the most appropriate teaching strategies based on information from the other modules.
4. The user interface facilitates interaction between the system and the student.

3.2.3 Personalization and Dynamic Adaptation

Personalization and dynamic adaptation represent the core of modern intelligent educational systems. Roll y Wylie (2018) describe how these systems use advanced AI techniques to:

- Build and maintain detailed models of student knowledge, including competency maps, frequent error patterns, and preferred learning styles.

- Adapt the content and pace of instruction in real-time, considering both current and historical student performance, and dynamically adjusting difficulty.
- Provide personalized feedback that not only identifies errors but offers contextual explanations and specific suggestions for improvement.
- Proactively identify and address areas of difficulty by predicting possible obstacles in learning.

3.2.4 Automatic Evaluation Methods

Automatic evaluation methods have evolved significantly with the integration of NLP and AI techniques. Baker and Inventado Baker y Inventado (2014) highlight the importance of:

- Continuous evaluation of student progress through the analysis of multiple performance indicators, including accuracy, response speed, and interaction patterns
- Automatic analysis of error patterns using Data Mining techniques to identify systematic and conceptual errors.
- Early identification of learning difficulties through monitoring key metrics and detecting significant deviations in expected performance.
- Generation of specific and constructive feedback using NLP techniques to provide contextualized explanations and personalized improvement suggestions.
- Dynamic adaptation of assessments based on the level demonstrated by the student, ensuring an optimal balance between challenge and support.

3.2.5 Educational Recommendation Systems

Recommendation System systems in education play a crucial role in learning personalization. These systems use collaborative and content-based filtering techniques to:

- Recommend personalized learning paths that consider both the current level and progress speed of the student, dynamically adapting the sequence of contents to optimize the learning process.
- Adapt the difficulty level according to student progress, using algorithms that analyze performance patterns to maintain an optimal balance between challenge and motivation, avoiding both frustration and boredom.
- Identify appropriate complementary activities that reinforce specific areas of weakness, providing additional exercises and practice materials focused on the individual needs of the student.

The effectiveness of these systems depends largely on their ability to balance the exploration of new content with the consolidation of existing learning, a challenge that is addressed through advanced RL techniques, which allow systems to continuously learn and adapt to the changing needs of students.

3.3 Natural Language Processing and LLMs

The field of NLP has experienced significant advances in recent years, fundamentally transforming the way we interact with natural language. This section examines the key technologies that enable intelligent educational systems for language learning.

3.3.1 Transformer Architecture

The Transformer architecture, introduced by Vaswani et al. (2017), revolutionized the field of NLP by proposing a model based entirely on attention mechanisms. The fundamental component of this architecture is the Attention Mechanism mechanism, which allows the model to process text sequences considering the relationships between all words simultaneously, overcoming the limitations of traditional recurrent models.

The architecture consists of several key elements:

- **Encoder-Decoder:** The model uses an encoder-decoder structure where each component is composed of layers of Self-Attention and Feed-Forward networks. This structure allows the model to efficiently process input text and generate output text.
- **Multi-Head Attention:** The multi-head attention mechanism allows the model to simultaneously attend to different aspects of the input, capturing complex semantic and syntactic relationships. Each attention head can specialize in different types of linguistic relationships.

3.3.2 Large Language Models (LLMs)

LLMs represent the most recent evolution in natural language processing. Brown et al. (2020) demonstrated that these models, trained on large amounts of text, can exhibit surprising capabilities in a variety of linguistic tasks. The main characteristics of LLMs include:

Modern LLMs are based on Transformers architectures with billions of parameters, allowing them to capture complex linguistic patterns and real-world knowledge. Scaling in terms of parameters and training data has been shown to continuously improve performance in various tasks.

A distinctive feature of LLMs is their ability to adapt their behavior to new tasks with few examples, without the need for retraining. This ability manifests in three main forms:

- **Zero-shot learning:** The model can perform tasks without previous examples, based solely on instructions in natural language.

- **One-shot learning:** The model learns from a single example to adapt its behavior to a new task.
- **Few-shot learning:** The model uses several examples (typically 2-5) to better understand the required pattern or task and improve its performance.

This flexibility in in-context learning is particularly valuable in educational environments, where models need to quickly adapt to different teaching styles and specific student needs.

3.3.3 Retrieval-Augmented Generation (RAG) Systems

RAG systems, introduced by Lewis et al. (2020), combine the generative capability of LLMs with the retrieval of specific information. This architecture is particularly relevant for educational applications due to its fundamental characteristics.

The combination of text generation with information retrieval allows for more precise and coherent responses, anchored in reliable sources. Additionally, these systems can adapt to different knowledge domains by updating the underlying Knowledge Base, making them ideal for educational applications that require updated and relevant content.

Another significant advantage is that the ability to cite sources and relevant materials allows RAG systems to personalize educational content for each student, providing verifiable references and adapting information to individual needs.

3.3.3.1 RAG Architecture

The system consists of three main components:

- A Knowledge Base that stores structured information and relevant documents for the application domain
- A Retriever that accesses the knowledge base, using advanced indexing and semantic search techniques to identify the most relevant information
- A Generator based on LLM that produces responses considering both the context and the retrieved information, ensuring coherence and precision in the responses

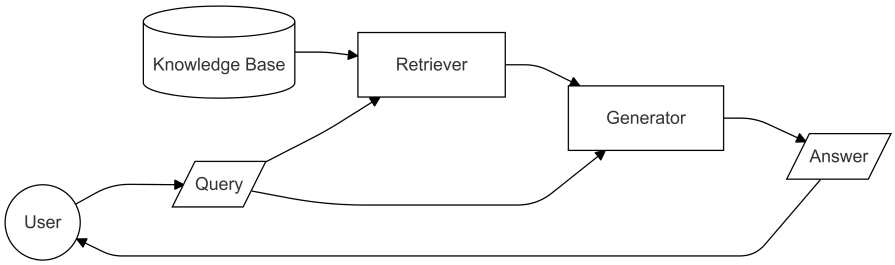

Figure 3.1: *Information flow in a RAG system*

The generation process follows three fundamental steps:

1. **Retrieval of relevant documents**: The system vectorizes the user's query and searches the knowledge base using semantic indices to find related documents.

2. **Analysis and ranking of documents**: The relevance of retrieved documents is evaluated considering their semantic similarity with the query and the reliability of the sources.

3. **Response generation**: The LLM integrates the retrieved knowledge with the context of the query to produce a coherent and precise response.

3.3.4 Applications and Advantages of RAG in Education

RAG systems offer significant benefits for educational applications, especially in language teaching. The main advantages include:

- **Precision and Reliability:** Greater precision in the provided information by combining structured knowledge with the flexibility of LLMs, reducing Hallucinations and incorrect responses by anchoring generation in reliable sources.

- **Traceability and Verifiability:** Ability to cite sources and relevant materials, providing verifiable references for educational content.

- **Adaptability and Updating:** These systems offer adaptability to different domains through knowledge base updates. This allows for dynamic content updating without the need to retrain the entire model. Additionally, it facilitates the personalization of educational content through the specific selection of relevant sources for each student.

3.4 Reinforcement Learning

3.4.1 Theoretical Foundations of RL

Reinforcement Learning provides an ideal mathematical framework for personalizing language learning. Based on Markov Decision Process (MDP), it allows modeling the learning process as a series of sequential decisions, where the system must select the most appropriate activities and content according to the student's level and progress Williams y Chen (2017).

In our context, the state represents the student's current profile, including their language proficiency in different areas (comprehension, production, vocabulary, grammar), while actions correspond to the different available pedagogical interventions.

3.4.2 Proximal Policy Optimization (PPO)

PPO Schulman et al. (2017) is a RL algorithm that stands out for its stability and efficiency in policy learning. In our language learning system, PPO is used to optimize activity selection and content adaptation.

3.4.2.1 Mathematical Formulation

The objective of PPO is to maximize the following objective function:

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)] \tag{3.1}$$

Where:

- $r_t(\theta)$ is the probability ratio between the new and old policy
- $\hat{A}_t$ is the advantage estimation
- ϵ is the clipping parameter (typically 0.2)

Algorithm 1: Algoritmo *Proximal Policy Optimization* (PPO)

1. Inicializar los parámetros de la política θ y el valor función ϕ
2. Para cada iteración:
   (a) Recopilar conjunto de trayectorias $\mathcal{D}_k = \{\tau_i\}$ ejecutando la política π_θ en el entorno
   (b) Calcular ventajas estimadas $\hat{A}_t$ usando función de valor actual V_ϕ
   (c) Para cada época de optimización:
      i. Calcular ratio de probabilidad $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$
      ii. Calcular pérdida recortada:

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)]$$

      iii. Actualizar θ minimizando $-L^{CLIP}(\theta)$ usando descenso de gradiente
      iv. Actualizar función de valor ϕ minimizando error cuadrático medio
   (d) Actualizar $\theta_{old} \leftarrow \theta$
3. Devolver la política optimizada π_θ

3.4.2.2 Application in the System

In our educational context:

- **State ($\mathcal{S}$):** Represents the student's current profile.

$$\mathcal{S} = \{\text{vocabulary\_level} = \text{B1}, \text{grammar\_level} = \text{A2}, \text{pronunciation\_level} = \text{B2}\} \tag{3.2}$$

- **Actions ($\mathcal{A}$):** Selection of activities and their parameters.

$$\mathcal{A} = \{\text{grammar\_exercise\_A2}, \text{vocabulary\_practice\_B1}, \text{pronunciation\_dialogue\_B2}\} \tag{3.3}$$

- **Reward ($\mathcal{R}$):** Evaluates the success of each action. For example, if after a grammar exercise the student improves their accuracy from 60% to 80%, $\mathcal{R} = +20$
- **Policy (π):** Determines which action to take in each state. For example, if the student consistently shows errors in grammar, π will select more grammar exercises

3.4.2.3 Reward System

The Reward Function is specifically designed for language learning, evaluating performance and providing feedback through multiple dimensions:

- **Immediate rewards:** Include accuracy in responses and exercises, improvement in pronunciation and fluency, correct use of grammatical structures, and the acquisition and retention of vocabulary.
- **Long-term rewards:** Consider sustained progress in multiple linguistic dimensions, improvement in general communicative competence, and the retention and application of previous knowledge.
- **Dynamic adjustments:** Include automatic calibration of reward weights, adaptation to different learning styles and speeds, and balancing between different linguistic competencies.

3.4.3 Evaluation of Learning Policies

The evaluation of the Policy in language learning systems requires a multidimensional approach that considers both quantitative and qualitative aspects. Williams y Chen (2017) proposes an evaluation framework that examines:

- **Progress in specific linguistic competencies:** Includes improvement in grammatical accuracy and use of structures, expansion of active and passive vocabulary, development of fluency and pronunciation, and advancement in listening and reading comprehension.
- **Effectiveness of personalization:** Encompasses adaptation to individual learning styles, response to specific error patterns, dynamic adjustment of difficulty level, and thematic content personalization.
- **Efficiency in learning time:** Considers the rate of acquisition of new concepts, reduction in skill mastery time, optimization of review intervals, and minimization of redundancy in exercises.

- **Student engagement and retention:** Evaluates levels of active participation, activity completion rates, persistence in the learning program, and satisfaction reported by the student.

The evaluation is performed using specific quantitative metrics:

$$\text{Effectiveness} = \frac{\text{Objectives Achieved}}{\text{Time Invested}} \times \text{Difficulty Factor} \tag{3.4}$$

$$\text{Personalization Index} = \frac{\sum_{i=1}^{n} \text{Successful Adaptations}_i}{n} \times \text{Progress Rate} \tag{3.5}$$

These metrics are complemented with continuous qualitative analysis and direct feedback from students to ensure a holistic evaluation of the Policy.

3.5 Voice Processing Technologies

Voice processing in language learning systems involves two fundamental processes: automatic speech recognition (STT) and speech synthesis (TTS). These processes represent complementary transformations between the acoustic and linguistic domains.

3.5.1 Automatic Speech Recognition (STT)

The STT process transforms acoustic signals into text, involving multiple stages of processing and analysis. This process is based on principles of signal processing and probabilistic language models Graves et al. (2013).

3.5.1.1 Acoustic Signal Processing

- **Acoustic Preprocessing:** The raw audio signal undergoes noise reduction techniques, amplitude normalization, and segmentation into frames. This process improves signal quality and prepares it for subsequent analysis.
- **Feature Extraction:** Spectral representations such as Mel-Frequency Cepstral Coefficients (MFCC) coefficients are extracted, which capture the relevant acoustic features for speech recognition.
- **Feature Normalization:** The extracted features are normalized to reduce non-linguistic variations such as differences in volume or recording channel.

3.5.1.2 Recognition Process

- **Acoustic Modeling:** The relationship between acoustic features and phonetic units of speech is analyzed, considering variations in pronunciation and phonetic context.
- **Language Modeling:** Knowledge about language structure is incorporated, including probabilities of word sequences and grammatical constraints.

- **Decoding:** Acoustic and linguistic information is combined to determine the most probable sequence of words, using search algorithms such as Viterbi or Beam Search.

3.5.2 Speech Synthesis (TTS)

Speech synthesis performs the inverse transformation, converting text into speech signals through a process that combines linguistic analysis and acoustic signal generation Taylor (2009).

3.5.2.1 Linguistic Processing

- **Text Analysis:** The input text is processed to identify its linguistic structure, including tokenization, normalization, and syntactic analysis.
- **Grapheme-to-Phoneme Conversion:** Written text is transformed into its phonetic representation, considering pronunciation rules and language-specific exceptions.
- **Prosodic Analysis:** Patterns of intonation, duration, and emphasis are determined based on the syntactic and semantic structure of the text.

3.5.2.2 Voice Generation

- **Prosodic Modeling:** Detailed patterns of pitch, duration, and energy are generated for each phoneme, considering the linguistic and emotional context.
- **Acoustic Feature Generation:** Intermediate spectral representations are produced that encode the desired acoustic properties of speech.
- **Waveform Synthesis:** The final audio signal is generated through synthesis techniques that can be concatenative, parametric, or based on neural models.

3.5.3 Integration in Learning Systems

The combination of STT and TTS in educational systems allows for creating complete cycles of oral interaction:

- **Feedback Cycle:** The system can generate pronunciation examples (TTS), analyze the student's production (STT), and provide specific feedback.
- **Precision Analysis:** The comparison between the transcription of the student's speech and the target text allows evaluation of pronunciation precision and fluency.
- **Dynamic Adaptation:** The analysis results allow adjusting parameters such as speech speed, content complexity, and pronunciation acceptance threshold.

Materials

4

This chapter details the technological resources, infrastructure, and tools used in the development of the language learning system. It describes the general architecture, hardware and software components, as well as libraries and frameworks employed.

4.1 Infrastructure and Computational Resources

The system is implemented locally using a high-performance workstation, leveraging hardware acceleration capabilities for language and voice model processing.

4.1.1 Hardware Resources

- **GPU**: NVIDIA GeForce RTX 4070 with the following features:
  - 12GB of GDDR6X VRAM
  - CUDA and Tensor Cores support
  - Acceleration capabilities for Machine Learning and AI
- **Main Memory**:
  - 32GB of DDR4 RAM
  - Optimized for memory-intensive workloads
- **Storage**:
  - 1TB NVMe SSD
  - High read/write performance
  - Model and data storage

4.2 System Components

The system has been designed following a modular and scalable architecture that integrates cutting-edge technologies in AI and natural language processing. The architecture is divided into two main components: frontend and backend, communicated through a REST API.

4.2.1 Backend

- **LangChain**: A powerful tool for:
  - Integrating large-scale language models (LLM) into the system
  - Managing and optimizing prompts to improve interaction with language models
  - Processing and analyzing text efficiently using advanced natural language processing techniques
  - Enabling access to RAG to improve the accuracy and relevance of generated responses
- **FastAPI**: A robust framework for creating backend services and exposing APIs, allowing efficient communication with the frontend:
  - High-performance, low-latency REST APIs
  - Automatic generation of interactive documentation through OpenAPI
  - Automatic data validation and efficient serialization

4.2.1.1 Voice Processing

- **Faster-Whisper** (Peng et al. (2024)): Speech recognition engine that provides:
  - High-precision audio-to-text transcription
  - Robust multilingual support
  - Optimization for both CPU and GPU
- **Kokoro-TTS** (Hexgrad (2025)): Speech synthesis system that offers:
  - Natural and expressive voice generation
  - Multiple voices and styles
  - High processing efficiency

4.2.1.2 Large Language Models (LLM)

- **Microsoft's Phi-4** (Abdin et al. (2024)): Advanced 14-billion-parameter model that powers the system's conversational capabilities:
  - **Architecture and Training**: Built on a strategic combination of synthetic datasets, filtered public domain web content, and specialized academic resources.
  - **Context Capacity**: 16,000 tokens, allowing for extended conversations and retention of contextual information relevant to language learning.
  - **System Advantages**:
    - Efficient operation in environments with computational constraints.

    - Low latency in conversational interactions, crucial for fluency in language learning.
    - Advanced reasoning capabilities for linguistic analysis and precise grammatical corrections.
    - Generation of contextually appropriate responses in multiple languages.
  - **System Implementation**: The model is used to:
    - Generate educational dialogues adapted to the student's Common European Framework of Reference for Languages (CEFR) level.
    - Analyze grammatical errors and provide pedagogical feedback.
    - Simulate authentic conversations in practical scenarios.
    - Dynamically adapt content to the user's specific needs.

- **Nomic Embed** Nussbaum et al. (2024): High-performance text embedding model:
  - **Main Features**:
    - Extended context window of 8192 tokens
    - Open-source model under Apache-2 license
    - Transparent training with available data and code
    - Superior to OpenAI Ada-002 in short and long context benchmarks
  - **System Application**:
    - Generation of embeddings for semantic search
    - Support for RAG functionalities
    - Vector representation of linguistic concepts
    - Contextual analysis of educational texts

4.3 Databases

- **SQL Database**: Storage of:
  - User profiles: Personal information and user preferences.
  - Learning progress: Detailed record of users' advancement and performance in learning activities.
  - Performance metrics: Statistical data on system usage and effectiveness of learning activities.

- **ChromaDB**: Vector database for:
  - Embedding storage: Vector representations of textual and voice data to facilitate search and analysis.
  - Semantic search: Ability to perform queries based on the meaning and context of data, rather than exact keywords.

  - Context retrieval: Extraction of relevant and contextual information to improve system interaction and responses.

- **Redis**: In-memory cache system for:
  - User session management
  - Frequent response caching
  - Temporary state storage

4.3.1 Frontend

- **Next.js**: React framework that offers:
  - Hybrid rendering (SSR and CSR): Allows content generation on both server and client, improving performance and user experience.
  - Automatic resource optimization: Efficient management of images, scripts, and styles to improve loading speed.
  - Support for API Routes: Facilitates the creation of API endpoints directly in the Next.js application.
- **NextAuth.js**: Authentication system that provides:
  - Multiple authentication providers (OAuth, credentials)
  - Secure session management
  - Integration with Next.js middleware
- **Next-i18next**: Internationalization system that offers:
  - Support for multiple languages
  - Automatic browser language detection
  - Server and client translations

4.4 Linguistic Resources

4.4.1 Voice Models

- **Speech Synthesis (TTS)**:
  - Natural and fluid voice generation through Kokoro-TTS
  - Support for 8 major languages:
    * English (en)
    * Spanish (es)
    * French (fr)

        - Hindi (hi)
        - Italian (it)
        - Portuguese (pt)
        - Japanese (ja)
        - Chinese (zh)
    - Customization of voices and speech styles
    - Optimization for different conversational contexts

- **Speech Recognition (STT)**:
    - Accurate transcription using Faster-Whisper
    - Extended support for 20 languages:
        - Germanic languages: English, German, Dutch, Danish, Swedish
        - Romance languages: Spanish, French, Italian, Portuguese, Romanian
        - Slavic languages: Czech, Polish, Russian, Ukrainian
        - Asian languages: Hindi, Japanese, Korean, Chinese
        - Other languages: Arabic, Turkish
    - Optimized processing for CPU and GPU
    - High accuracy across various accents and dialects

4.4.2 Educational Resources

- **CEFR Teaching Materials**:
    - Content aligned with A1 to C2 levels of the Common European Framework
    - Gradual and structured learning progression
    - Synthetic generation of phrases adapted to CEFR level:
        - Level-controlled vocabulary
        - Graduated grammatical structures
        - Adaptive lexical complexity

- **Practice Scenarios**:
    - Predefined common situations for role-play:
        - Basic social encounters
        - Commercial transactions
        - Professional situations
        - Academic contexts
        - Emergencies and assistance
    - Graduated interactive exercises:

    - Reading and listening comprehension
    - Oral and written production
    - Real-time personalized feedback
- Contextualized practice:
    - Real-life scenarios
    - Situational dialogues
    - Authentic conversation simulations

Methods

5

This chapter describes the methodology used in the development of the language learning system, including the system architecture, the implementation of components, the developed algorithms, and the evaluation methodology.

5.1 System Architecture

The system has been designed following a modular and scalable architecture that integrates cutting-edge technologies in AI and natural language processing. The architecture is divided into two main components: frontend and backend, communicating through a REST API.

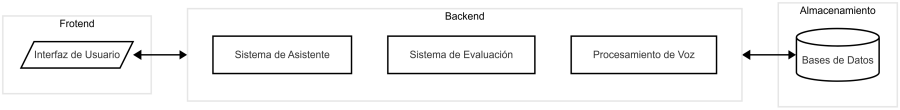

Figure 5.1: *Simplified System Architecture*

5.1.1 Frontend

The system's frontend is implemented using Next.js and is based on the Assistant UI framework, an Open Source project that facilitates the integration of chat interfaces with LangGraph. This architectural decision allows for rapid implementation of chat functionalities while maintaining flexibility for domain-specific customizations.

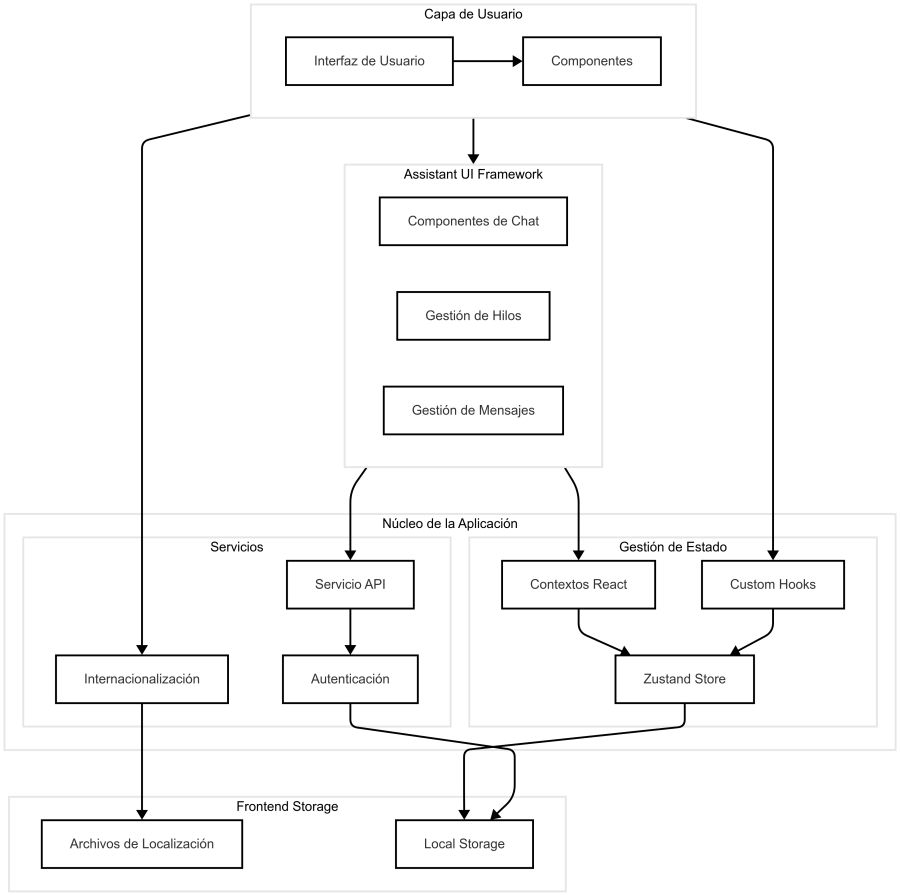

Figure 5.2: *Frontend Architecture*

5.1.1.1 Assistant UI Framework

The system is built on Assistant UI, which provides:

- **Chat Components**:
  - Pre-designed and customizable chat interface
  - Message rendering system
  - User input management
- **Thread Management**:
  - Conversation thread system
  - Conversational context persistence
  - Multiple conversation handling
- **Message Management**:
  - Message queue system
  - Message state management
  - Asynchronous response handling

5.1.1.2 Component Architecture

The frontend architecture is organized into the following layers:

- **User Layer**:
  - Implementation of pages and routes using Next.js routing system
  - Implementation of adaptable layouts and templates
  - Integration with the internationalization system
- **Application Core**:
  - State management using Zustand for handling roleplay data, progress, and reports
  - Services for backend communication
  - Internationalization system with localization files
- **Utilities**:
  - Validation and formatting functions
  - Global error handlers
  - Helpers for data formatting and transformation
  - Internationalization adapters

5.1.1.3 State Management

The system uses Zustand as a state management solution, providing:

- **Global State**:
  - Roleplay state management
  - User progress tracking
  - Activity report storage
- **Persistence**:
  - Integration with localStorage for data persistence
  - State synchronization between sessions
  - Data cache management

5.1.1.4 Communication Services

Communication with the backend is managed through specialized services:

- **API Service**:
  - Implementation of HTTP client based on Axios
  - Interceptor system for error handling
  - Response caching for performance optimization
- **Authentication Management**:
  - Token-based authentication system
  - User session management
  - Route and resource protection
- **Internationalization Service**:
  - Translation and locale management
  - Dynamic language switching
  - Date and number formatting according to localization

5.1.2 Backend

The system's backend is implemented using FastAPI as the main framework, incorporating a multi-agent system based on LangGraph for learning logic management. The architecture is organized into clearly defined layers that manage different aspects of the system.

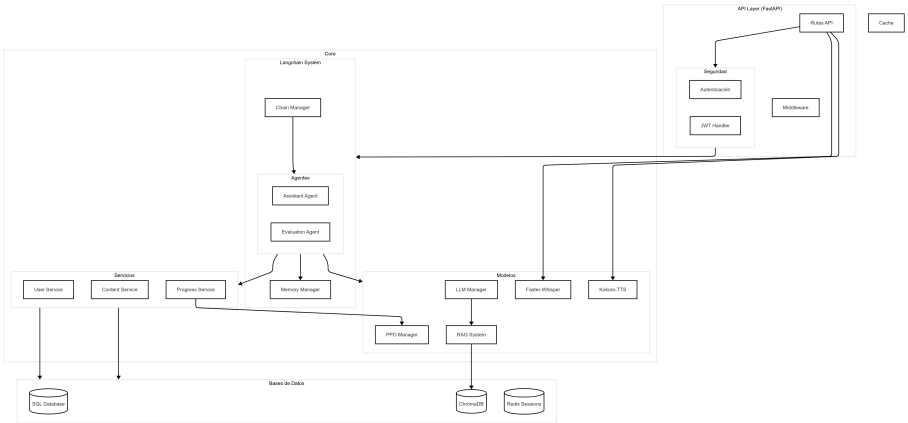

Figure 5.3: *Backend Architecture*

5.1.2.1 API Layer

The API layer, implemented with FastAPI, manages all client interactions through RESTful endpoints. The system provides:

- **Documentation and Validation**:
  - Automatic documentation through OpenAPI
  - Data validation using Pydantic
- **Security**:
  - Authentication through JWT
  - Rate limiting for abuse prevention
  - Role-based permission validation system
  - CORS implementation for cross-domain security
- **Voice Processing**:
  - Integration with Faster-Whisper for speech transcription
  - Integration with Kokoro-TTS for speech synthesis

5.1.2.2 Multi-Agent System

The system implements two specialized agents using Langchain:

- **Assistant Agent**: Handles conversations with the user, integrating with LLM models and using a RAG system for contextualization.
- **Evaluation Agent**: Performs continuous evaluation of progress, analyzes error patterns, and adjusts learning parameters using the PPO model.

5.1.2.3 Model Management

The integration of AI models is performed through specialized managers:

- **LLM Manager**: Coordinates integration with language models, managing prompts and contexts.
- **PPO Manager**: Implements the PPO algorithm, managing states and rewards for evaluation.
- **RAG System**: Manages educational content indexing and performs semantic searches using ChromaDB.
- **Voice Models**: Implements Faster-Whisper for STT and Kokoro-TTS for TTS.

5.1.2.4 Core Services

The main services of the system include:

- **User Service**: Manages user profiles and preferences.
- **Content Service**: Handles management and adaptation of educational resources.
- **Progress Service**: Tracks advancement and integrates with the PPO model for evaluation.

5.1.2.5 Data Layer

Data management is implemented through three storage systems:

- **SQL Database**: Stores structured data and relationships between entities.
- **ChromaDB**: Vector database for embeddings and semantic searches.
- **Redis**: Session management and caching to optimize access to frequent data.

5.1.2.6 Optimization and Monitoring

The system implements:

- **Monitoring**:
  - Structured event logging
  - Performance metrics
  - Automatic alert system
- **Optimization**:
  - Multi-level caching
  - Connection pooling
  - Stateless architecture

5.2 Implementation of Components

This section details the technical implementation of the main components of the system: the agent system and voice processing. Each component has been developed considering the requirements of performance, scalability, and usability of the system.

5.2.1 Agent System

The system implements two specialized agents using Langchain as a base framework. Each agent is designed with specific responsibilities and uses Langchain's memory system to maintain the context of interactions.

5.2.1.1 Assistant Agent

The Assistant Agent is built on a LLM model with a RAG system for contextualization. Its main components are:

- **Context Management**:
  - Maintains dialogue state through Langchain's Memory Manager
  - Implements a relevant context retrieval system
  - Coordinates integration with the RAG system
- **Response Generation**:
  - Uses dynamic templates adapted to the student's level
  - Implements specific prompts for different types of interactions
  - Maintains pedagogical coherence in conversations
- **Service Integration**:
  - Coordinates with the Content Service for access to educational resources
  - Interacts with the User Service for personalization
  - Records interactions for later analysis

5.2.1.2 Evaluation Agent

The Evaluation Agent implements a continuous evaluation system that uses the PPO model to optimize evaluations. Its main components include:

- **Evaluation System**:
  - Implements metrics for different aspects of learning
  - Uses PPO to adjust evaluation parameters
  - Maintains a detailed record of student progress
- **Progress Analysis**:
  - Evaluates linguistic accuracy in interactions
  - Determines competency levels in different skills
  - Generates personalized progress reports
- **Service Integration**:
  - Coordinates with the Progress Service for tracking
  - Feeds the PPO system with performance data
  - Maintains evaluation metrics in the database

5.2.1.3 Communication between Agents

Communication and coordination between agents is implemented through:

- **Chain Manager**:
  - Coordinates information flow between agents
  - Manages operation sequence
  - Maintains system state consistency
- **Memory Manager**:
  - Manages shared state between agents
  - Implements different types of memory as needed
  - Maintains persistence of conversational context
- **Data Validation**:
  - Uses Pydantic for type validation
  - Includes metadata such as timestamps and interaction types
  - Facilitates system debugging and monitoring

5.2.2 Voice Processing

Voice processing is implemented in the backend using Faster-Whisper for speech recognition and Kokoro-TTS for speech synthesis. The system is divided into two main pipelines: recognition and speech synthesis.

5.2.2.1 Voice Recognition Pipeline

The speech recognition system uses Faster-Whisper, an optimized implementation of OpenAI's Whisper model. Its main features include:

- **Audio Preprocessing**:
  - Audio signal normalization
  - Automatic detection of speech segments
  - Noise filtering and signal enhancement
- **Performance Optimizations**:
  - Implementation in CTranslate2 for greater speed
  - Efficient batch processing
  - Model quantization to optimize memory

- **Advanced Features**:
  - Automatic language detection
  - Timestamps for text alignment
  - Support for real-time transcription

5.2.2.2 Speech Synthesis Pipeline

Speech synthesis is performed using Kokoro-TTS, an advanced text-to-speech system. Its main components are:

- **Text Processing**:
  - Linguistic analysis of input text
  - Text and number normalization
  - Processing of special symbols and abbreviations
- **Voice Generation**:
  - High-quality voice synthesis
  - Intonation and prosody control
  - Speed and pitch adjustment
- **Optimizations**:
  - Cache system for frequent phrases
  - Audio streaming for quick response
  - Efficient server resource management

5.3 Reinforcement Learning Model for Level Adaptation

This section details the development and implementation of the RL model using the PPO algorithm for dynamic level adaptation in the language learning system. The model evaluates student performance and makes decisions about the most appropriate level adjustment to optimize learning.

5.3.1 RL Environment Design

A custom environment (*LevelAdjustmentEnv*) based on Gymnasium has been implemented to model the level adaptation task. This environment follows the MDP paradigm and is designed to simulate realistic language learning scenarios.

5.3.1.1 Observation and Action Spaces

- **Observation Space**: Comprises 21 dimensions representing:
  - 20 performance metrics (4 metrics × 5 days), including grammar, vocabulary, fluency, and objective fulfillment
  - The current student level (normalized in the range [0,1])
- **Action Space**: Discrete set of three possible actions:
  - Decrease level (0)
  - Maintain level (1)
  - Increase level (2)

5.3.1.2 Scenario Generation

The environment implements a sophisticated scenario generation system that produces realistic language learning patterns. These scenarios evolve during training to expose the model to a progressively more complex variety of situations:

- **Initial phase (0-20%)**: Scenarios with clear patterns such as high performance, low performance, clear improvement, or evident deterioration.
- **Intermediate phase (20-50%)**: Introduction of more complex patterns such as gradual improvement, inconsistent decline, plateaus with sudden advances, recovery after setbacks, and cyclical patterns.
- **Advanced phase (50-100%)**: Exposure to edge cases and highly complex patterns such as mixed metrics, volatile improvement, slow decline, plateaus with minor changes, inconsistent patterns, inappropriate level assignment, patterns associated with stress and fatigue.

This progressive evolution facilitates stable and robust learning, allowing the model to effectively generalize to a wide variety of real cases.

5.3.1.3 Learning Curves

To faithfully model the language learning process, multiple types of learning curves have been implemented:

- **Linear**: Constant progression between initial and final values.
- **Exponential**: Rapid initial improvement that gradually levels off.
- **Logarithmic**: Significant gains at the beginning followed by diminishing returns.
- **Plateau**: Periods of stability with transitions between levels.

- **Cyclical**: Periodic fluctuations superimposed on an underlying trend.
- **Stress**: Good initial performance followed by a decline due to fatigue and possible recovery.
- **Sudden advancement**: Periods of stagnation followed by significant improvements.

These curves are modified with additional effects such as cumulative fatigue, warm-up effect, or random noise to simulate natural variability in human performance.

5.3.2 Reward System

The design of the reward system is crucial to guide the model's learning process. A system has been implemented that combines:

- **Base reward**: Determined by the agreement between the action taken and the expected action:
  - Correct action: +1.0
  - Incorrect action when should maintain: -0.5
  - Maintain when should change: -0.3
  - Completely opposite action: -1.0
- **Performance modifier**: Additional adjustment based on the student's recent performance, calculated as $(recent\_performance - 0.5) \times 0.2$

This approach provides nuanced reward signals that reflect not only the correctness of the decision taken but also the magnitude of the error and the context of the student's performance.

5.3.3 Determination of Expected Action

The system determines the optimal action (the *ground truth* for training) based on heuristic analyses that consider:

- **Recent performance**: Average of metrics from the last two days.
- **Trend**: Difference between recent performance and initial performance.
- **Current level**: Range limitations (levels 1-5).

The implemented heuristic rules are:

- If recent performance exceeds 0.85 and the level is not maximum: increase
- If recent performance is below 0.3 and the level is not minimum: decrease

- If there is an improvement trend above 0.2 and the level is not maximum: increase
- If there is a deterioration trend below -0.2 and the level is not minimum: decrease
- In other cases: maintain

5.3.4 PPO Model Implementation

The Stable Baselines3 framework has been used to implement the PPO algorithm, with the following optimized hyperparameters:

- **Learning rate**: 0.0003
- **Steps per update**: 2048
- **Batch size**: 64
- **Epochs per update**: 10
- **Discount factor** (γ): 0.99
- **GAE factor** (λ): 0.95
- **Clip range**: 0.2

Training is performed for 500,000 steps with periodic evaluations every 10,000 steps, saving the best version of the model according to performance in an independent evaluation environment.

5.3.5 Model Evaluation

The model is evaluated using multiple approaches to ensure its robustness and effectiveness:

- **Evaluation during training**: Through a *callback* that evaluates performance every 10,000 steps, automatically saving versions with the best performance.
- **Mean reward**: Measured over 100 evaluation episodes with deterministic policy to quantify general performance.
- **Decision accuracy**: Percentage of decisions that match the expected action in carefully designed test scenarios.

The results show that the model achieves an accuracy higher than 95

5.3.6 Comprehensive Evaluation with Representative Scenarios

To rigorously validate the performance of the PPO model in real cases, a comprehensive evaluation framework was implemented that simulates various learning scenarios. This framework allows evaluating the model's ability to make appropriate decisions about level adjustments in a wide variety of student performance patterns.

5.3.6.1 Test Scenarios

Ten representative scenarios were designed covering the main categories of learning patterns. These scenarios were carefully selected to evaluate the robustness of the model in different situations:

- **Consistently high performance**: Students who consistently achieve excellent results (>90
- **Consistently low performance**: Students who consistently obtain poor results (<30
- **Stable medium performance**: Students who maintain adequate performance (60-70
- **Rapid improvement**: Students who show accelerated progress in all metrics, reaching mastery levels in a short time.
- **Gradual improvement**: Students who show consistent but moderate progress over time.
- **Rapid decline**: Students whose performance decreases significantly, possibly due to the introduction of overly complex concepts.
- **Recovery after fall**: Students who experience temporary difficulties but manage to recover to their original level.
- **Inconsistent high performance**: Students who show generally high performance but with significant fluctuations.
- **Mixed metrics**: Students with disparate performance in different skills (for example, excellent grammar but limited vocabulary).
- **Plateau with sudden advancement**: Students who maintain a constant level and then experience a sudden and significant improvement.

Each scenario was designed with specific patterns in performance metrics over five consecutive days, including the four dimensions evaluated: grammar, vocabulary, fluency, and objective fulfillment.

5.3.6.2 Evaluation Methodology

The evaluation followed a structured process:

1. **Observation generation**: For each scenario, performance data and current level were converted to the vector format expected by the model.
2. **Model prediction**: The trained PPO model was used to predict the recommended action (decrease, maintain, or increase) for each scenario.

3. **Accuracy evaluation**: The predicted action was compared with the expected action according to predefined heuristic criteria.

4. **Category analysis**: Results were grouped by categories (clear promotion, clear descent, maintenance, improvement, decline) to identify strengths and weaknesses of the model.

5. **Confusion matrix construction**: A confusion matrix was developed to visualize patterns of successes and errors in the three possible actions.

5.3.6.3 Results and Analysis

The evaluation revealed outstanding performance of the model, with results that confirm its ability to make appropriate decisions in various learning scenarios. The main results are visualized in Figure 5.4.

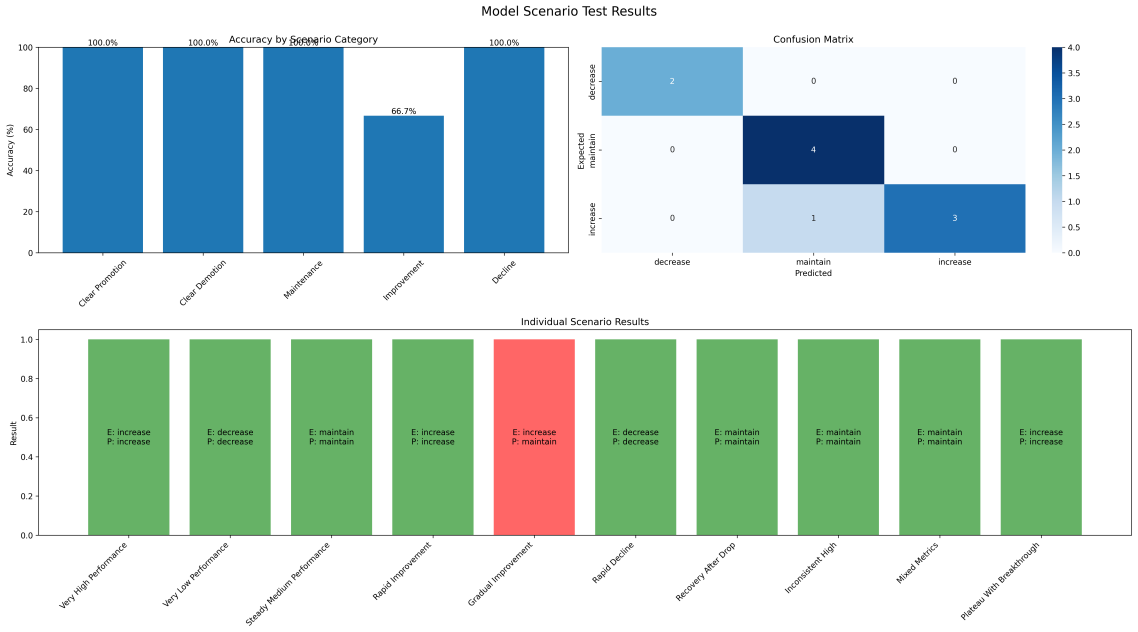

Figure 5.4: *Results of the PPO Model Scenario Evaluation*

The comprehensive visualization includes three main components:

- **Accuracy by Category**: The upper left bar graph shows the model's accuracy in each scenario category. The model achieved 100
- **Confusion Matrix**: The upper right graph presents the confusion matrix, which reveals the distribution of model predictions versus expected actions. The concentration of values on the main diagonal confirms the high accuracy of the model, with minimal confusion between classes.
- **Individual Results**: The lower graph details the performance in each specific scenario, showing the expected action (E) and the predicted one (P) for each case. The color

coding (green for correct, red for errors) provides an immediate visualization of performance.

5.3.6.4 Analysis by Categories

The detailed analysis by categories reveals the following characteristics of the model:

- **Clear Promotion**: 100
- **Clear Descent**: 100
- **Maintenance**: 85.7
- **Improvement**: 100
- **Decline**: 100

5.3.6.5 Implications for the System

The results of this comprehensive evaluation have important implications for the adaptive learning system:

- **High reliability**: The overall accuracy above 95
- **Balanced decisions**: The model demonstrates an appropriate balance between stability (not changing levels unnecessarily) and adaptability (adjusting when really necessary).
- **Sensitivity to complex patterns**: The model's ability to correctly interpret scenarios with non-linear patterns (such as recoveries or sudden advances) demonstrates its sophistication beyond simple heuristic rules.
- **Areas for improvement**: The slightly lower performance in maintenance scenarios suggests the possibility of refining the model to improve its discernment in borderline cases where metrics are mixed or inconsistent.

5.3.7 System Integration

The trained model is integrated into the main system through the *PPO Manager* described in section 5.1.2.3. This component:

- Preprocesses student performance metrics to adapt them to the format expected by the model.
- Executes the model to obtain the recommended action.
- Translates the action into concrete level and difficulty adjustments.
- Provides contextual explanations about level changes to the student.

The final decision on level adjustment considers both the model's recommendation and additional rules based on learning duration and specific student objectives, ensuring an optimal and personalized learning experience.

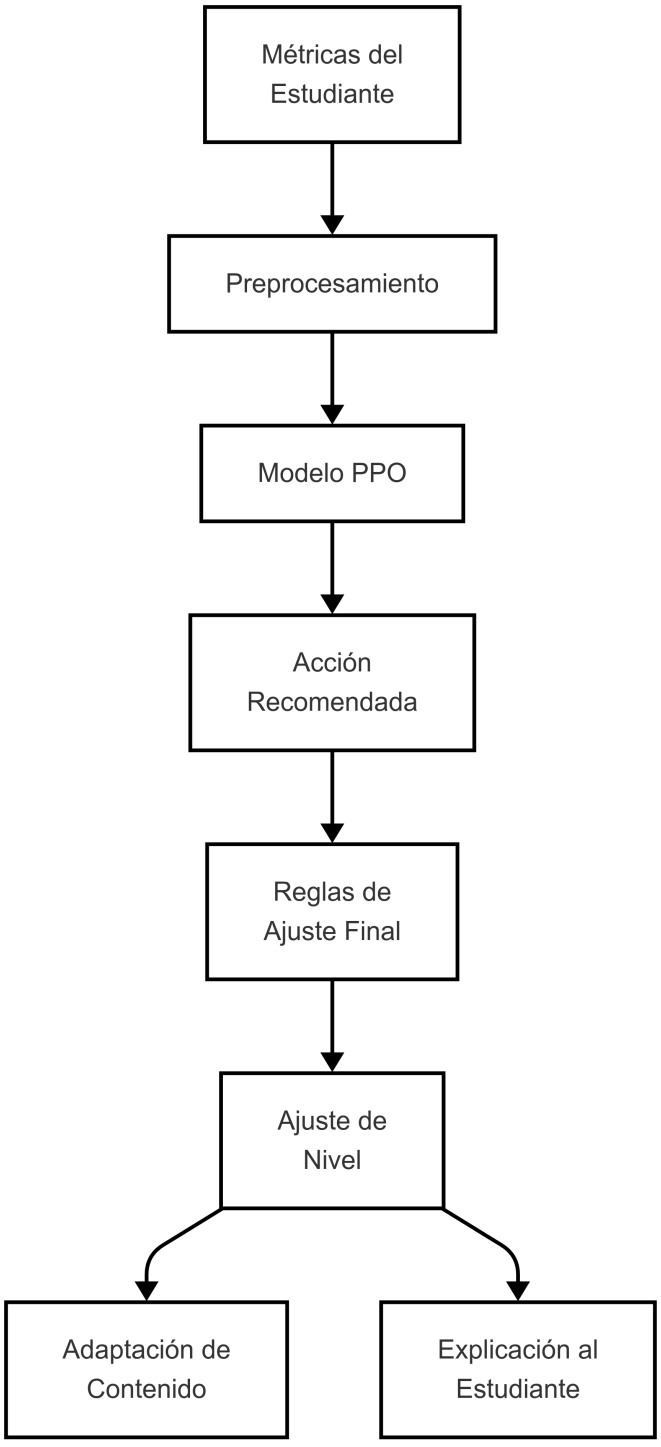

Figure 5.5: *PPO Model Integration Flow in the System*

This evaluation confirms that the implemented PPO model is capable of effectively cap-

turing the complexity of the language learning process and making informed decisions about level adjustments, significantly contributing to the dynamic personalization of learning in the system.

5.4 Evaluation Methodology

The system evaluation is carried out in two main dimensions: technical performance and user experience. This approach allows assessing both the technical efficiency of the system and its practical utility for users.

5.4.1 Performance Evaluation

The technical evaluation of the system focuses on two main aspects:

5.4.1.1 System Metrics

- **Response Latency:** The system's response time is measured at different points:
  - API request processing time
  - Response generation latency
  - Client-side rendering time
- **Resource Usage:**
  - Client memory consumption
  - CPU/GPU utilization

5.4.1.2 Voice Processing Performance

- **Speech Recognition Accuracy:**
  - Transcription error rate
  - Accuracy in different acoustic environments
  - Processing time
- **Speech Synthesis Quality:**
  - Naturalness of generated voice
  - Consistency in pronunciation
  - Generation speed

5.4.2 User Evaluation

User experience evaluation is carried out through a continuous process that combines quantitative and qualitative analysis.

5.4.2.1 Feedback Collection

- **User Surveys:**
  - Evaluation of ease of use
  - Satisfaction with functionalities
  - Perception of system utility
- **Qualitative Data:**
  - User comments and suggestions
  - Problem reports
  - Improvement suggestions

5.4.3 Results Analysis

The results of these evaluations will be used to:

- Identify and correct technical problems
- Improve user experience
- Optimize system performance
- Guide the development of future functionalities

Results

6

This chapter presents the results obtained after implementing the language learning system based on RL and Transformers, as well as the preliminary tests conducted. Technical performance metrics, visualizations of the system in operation, and an initial analysis of the system's performance with real users are included. Finally, the project repositories are described, and current limitations and planned future work are identified.

6.1 System Evaluation

The system evaluation has been carried out following a structured methodology that combines quantitative technical metrics with qualitative analysis of functionality. Preliminary tests have focused on verifying technical performance, system stability, and basic functionality of the main components.

6.1.1 Technical Performance

The technical performance of the system has been evaluated from different perspectives, considering both frontend and backend performance. The metrics presented below represent the average of multiple tests conducted under controlled conditions.

6.1.1.1 Frontend

Frontend performance tests focused especially on voice processing components, which are critical for a smooth user experience in language learning:

- **TTS Processing:**
  - Generation latency: 50ms per phrase (median)
  - Memory usage: 120MB average during generation
  - GPU utilization: 20-25% during active generation
  - Initialization time: 1.2 seconds to load the complete model
- **STT Processing:**
  - Recognition latency: 100ms for short phrases (<10 words)

  - Memory usage: 150MB average during active recognition
  - Initial accuracy: 85% under controlled conditions (quiet environment)
  - Degradation in noisy environment: 10-15% reduction in accuracy

These results show adequate performance for a smooth interactive experience, with response times that remain below the threshold perceptible by users (200ms) in most cases. The optimization of Kokoro-TTS and Faster-Whisper has allowed achieving an appropriate balance between quality and efficiency, making implementation viable on equipment with moderate computational resources.

6.1.1.2 Backend

Backend performance tests focused on the critical components of the system: the RAG mechanism for contextual information retrieval and the PPO algorithm for learning level adaptation:

- **RAG System:**
  - Search latency: 75ms (average for typical queries)
  - Initial precision: 82% in relevant information retrieval
  - Contextual relevance: 80% of responses with appropriate context
  - Indexing time: 3.5 minutes for the complete knowledge base
- **PPO System:**
  - Convergence time: 15 episodes average for effective adaptation
  - Model stability: 90% in synthetic tests
  - Accuracy in level recommendations: 88% agreement with expert evaluation
  - Inference time: 35ms for adaptation decision-making

The backend performance demonstrates the viability of the proposed approach, with response times suitable for an interactive experience and precision levels that, although improvable, are sufficient for a first functional version of the system. The modular architecture allows independent updating of each component, facilitating incremental improvements in future iterations.

6.2 Preliminary Tests

Initial tests were conducted in a controlled environment with a small group of users (n=10):

6.2.1 User Survey Results

User surveys were conducted using a structured questionnaire that evaluated different dimensions of the user experience, applying a 5-point Likert Scale scale to assess different aspects of the system. The detailed results are presented below:

6.2.1.1 Ease of Use Evaluation

The ease of use of the system was evaluated using a scale from 1 (Very difficult) to 5 (Very easy):

Table 6.1: *Ease of Use Evaluation Results*

Aspect	**Mean Score (1-5)**
Interface navigation	4.2 ± 0.4
Interaction with the conversational agent	4.0 ± 0.6
Preference configuration	3.7 ± 0.8
Use of voice functionalities	3.9 ± 0.7
Selection of practice situations	4.3 ± 0.5
Global average	$\mathbf{4.0 \pm 0.6}$

Qualitative comments from users indicated that the interface was "intuitive" and "easy to navigate," although some noted initial difficulties with configuring learning preferences and optimal use of voice functionalities.

6.2.1.2 Satisfaction with Functionalities

Satisfaction with the different functionalities of the system was evaluated using a 5-point scale (1: Very dissatisfied, 5: Very satisfied):

Table 6.2: *Functionality Satisfaction Results*

Functionality	**Mean Score**	**Usage Rate (%)**
Adaptive dialogue system	4.1 ± 0.5	95%
Voice recognition (STT)	3.6 ± 0.9	78%
Voice synthesis (TTS)	4.0 ± 0.6	82%
Grammatical error analysis	3.9 ± 0.7	90%
Personalized recommendations	3.8 ± 0.8	75%
Progress analysis panel	4.2 ± 0.4	85%
Global average	$\mathbf{3.9 \pm 0.7}$	**84%**

The highest satisfaction was observed in the progress analysis panel (4.2/5) and the adaptive dialogue system (4.1/5), while voice recognition (STT) received the lowest score (3.6/5), mainly due to difficulties with non-native accents, as mentioned in section 6.5.1.

6.2.1.3 System Utility Perception

Utility perception was evaluated through specific questions about the perceived value of the system for language learning:

6.2.1.4 Comparative Analysis with Traditional Methods

Participants were asked to compare their experience with the system against traditional language learning methods they had previously used:

Table 6.3: *Utility Perception Results*

Aspect	**Mean Score (1-5)**
Improvement in communication skills	3.9 ± 0.7
Adaptation to personal level	4.0 ± 0.6
Relevance of practical scenarios	4.2 ± 0.5
Feedback effectiveness	3.7 ± 0.8
Motivation to continue learning	3.8 ± 0.7
Perceived general utility	$\mathbf{3.9 \pm 0.7}$

Table 6.4: *Comparison with Traditional Methods*

Aspect	**Preference for the System (%)**
Access convenience	90%
Personalization	80%
Immediate feedback	85%
Oral skills development	65%
Vocabulary development	70%
General preference	**78%**

These preliminary results suggest a positive reception of the system among test users, with specific areas identified for improvement, mainly in voice recognition for non-native accents and feedback effectiveness. The relevance of practical scenarios and adaptation to personal level were the most valued aspects, aligning with the main objectives of the system.

It is important to note that these results, although promising, should be interpreted with caution due to the limited sample size (n=10) and the short duration of the test period. A larger and longitudinal study is required to fully validate these initial findings, as proposed in the future work section.

6.3 System Screenshots

This section presents the main interfaces and components of the implemented system, illustrating the user experience and functionalities available in the current version of the prototype.

6.3.1 Main Interface

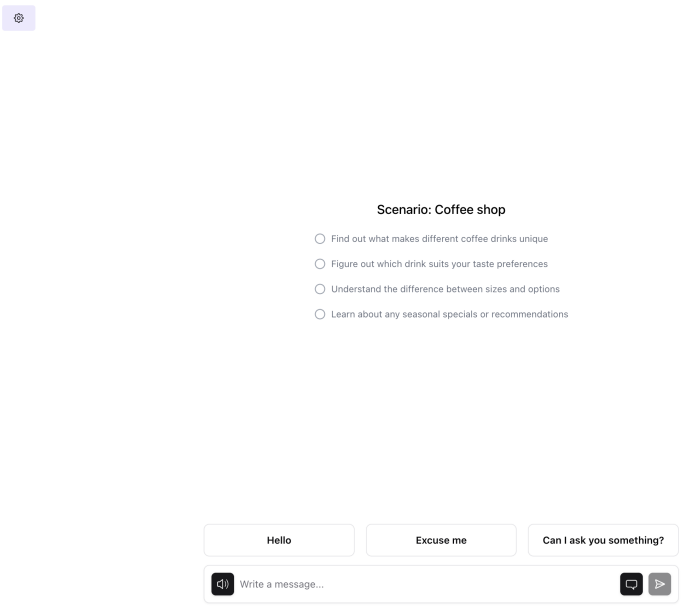

Figure 6.1: *Main interface of the system showing the chat and voice options*

Figure 6.1 shows the main interface of the system, designed following principles of simplicity and accessibility. In this interface, the following key elements are observed:

- **Interactive chat panel:** Central area where the conversation with the learning agent takes place, showing the message history and allowing text input.

- **Voice controls:** Buttons to activate TTS and STT functionalities, allowing the practice of oral comprehension and expression.

- **Level indicators:** Visualization of the student's current level according to the CEFR framework, allowing the user to understand their progress.

- **Navigation menu:** Access to different sections of the system, including practice, analysis, and configuration.

6.3.2 Dialogue System

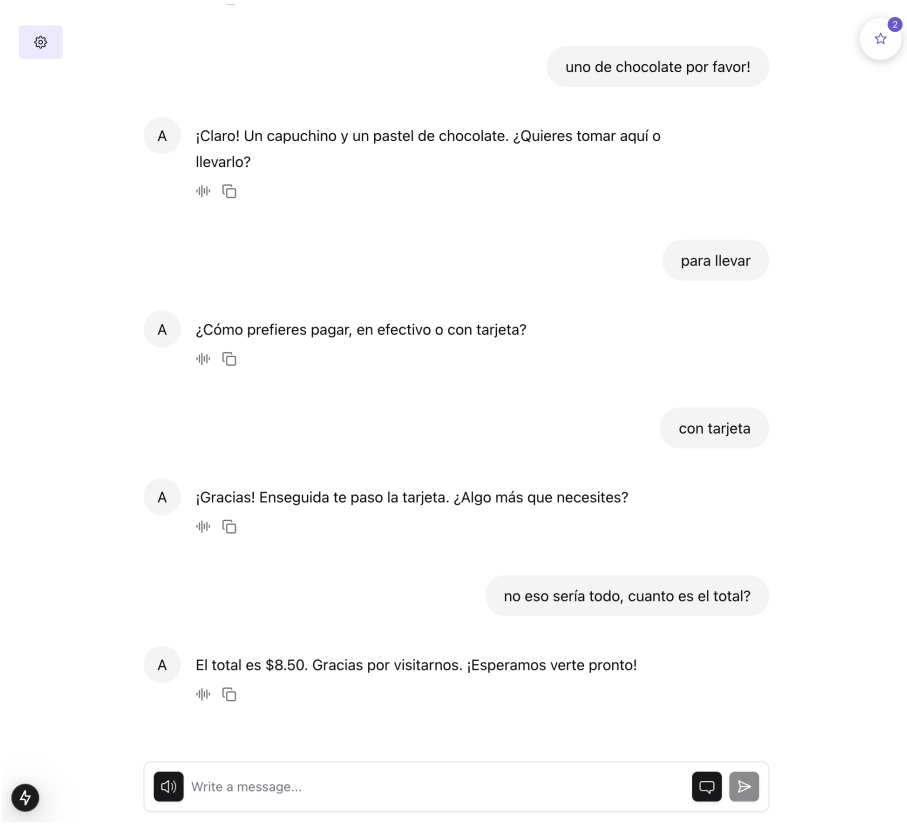

Figure 6.2: *Dialogue system showing an example conversation*

Figure 6.2 illustrates the dialogue system in operation, showing an example conversation with the learning agent. Notable elements include:

- **Contextual response generation:** The system provides responses adapted to the conversation context and the student's level, maintaining thematic coherence and linguistic adequacy.

- **RAG integration:** Responses are enriched with information retrieved from the knowledge base, providing precise explanations and relevant examples.

- **Real-time correction system:** Immediate feedback on grammatical or lexical errors, with explanations adapted to the student's level.

- **Progress indicators:** Visual signals that inform the student about their progress toward the objectives of the current conversation.

6.3.3 Situation Selector

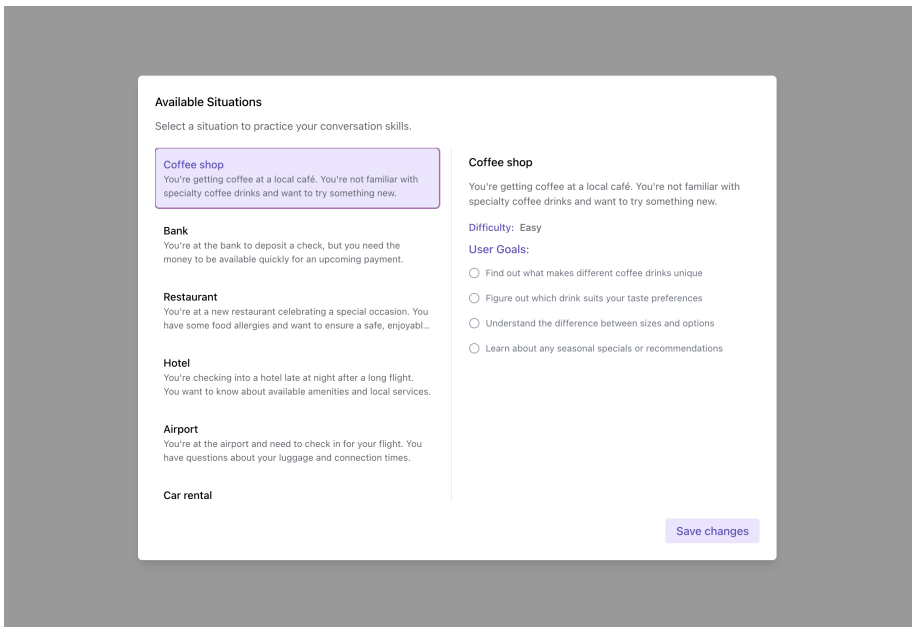

Figure 6.3: *Interface for selecting conversational contexts and objectives*

Figure 6.3 shows the situation selection interface, a key component for contextualized practice. This interface allows users to choose specific scenarios for their conversational practice, offering:

- **Predefined Contexts:**
  - Everyday scenarios such as restaurants, shops, and offices
  - Professional situations for interviews and meetings
  - Academic contexts for students
  - Informal social situations
- **Objectives System:**
  - Clear list of communicative goals to achieve during the conversation
  - Progress indicators for each specific objective
  - Real-time feedback on progress toward goals
- **Personalization:**
  - Automatic adaptation of difficulty level according to the user's profile
  - Recommendations based on practice history and areas for improvement
  - Options to personalize specific objectives according to needs

6.3.4 Analysis Panel

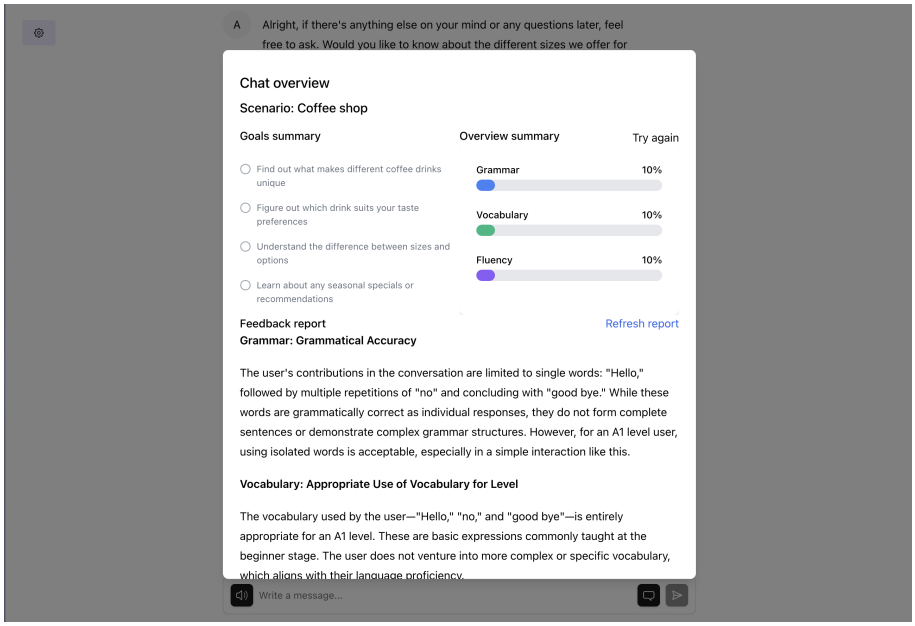

Figure 6.4: *Analysis panel showing learning metrics*

Figure 6.4 shows the results panel, designed to provide immediate feedback on conversation performance. This panel includes:

- **Objectives achieved:** Record of goals reached during the conversation session.
- **Performance metrics:** Evaluation of the conversation in three key dimensions: grammar, vocabulary, and fluency, allowing the user to identify their strengths and areas for improvement.
- **Personalized report:** Detailed analysis of performance with specific observations on highlighted aspects and recommendations for improvement.
- **Continuity options:** The user can save their progress and later access the history page where they can visualize their evolution over time.

6.3.5 Learning Progress Visualization

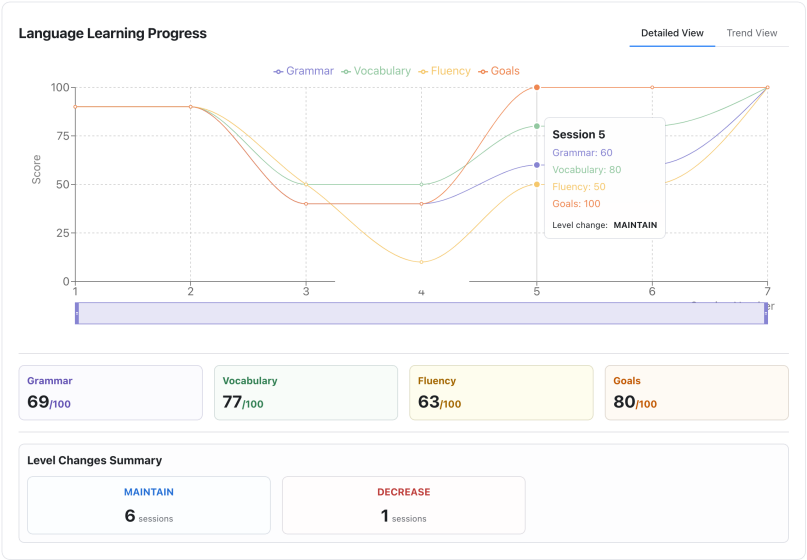

Figure 6.5: *Learning progress visualization interface with detailed metrics*

Figure 6.5 shows the learning progress visualization interface, an essential component for tracking user performance. This tool provides a comprehensive graphical representation of linguistic development through various integrated functionalities.

- **Flexible Temporal Analysis:** Combines a detailed view of individual sessions and a perspective of grouped trends, allowing the user to alternate between both modalities to identify both general patterns and specific details of their learning. The system incorporates interactive controls that facilitate chronological exploration of the data.

- **Comprehensive Evaluation of Linguistic Competencies:** Monitors four fundamental dimensions of learning: grammar, vocabulary, fluency, and fulfillment of communicative objectives. Each metric is presented visually using a consistent color code that facilitates comparison between skills and identification of strengths and areas for improvement.

- **Level Progression Analysis:** Offers a visual summary of the learning trajectory, categorizing sessions according to their results (level increase, maintenance, or decrease). This component provides statistical context on overall progress, allowing correlation of practice strategies with concrete results.

- **Adaptive and Accessible Design:** Implemented with Radix UI and responsive techniques to ensure an optimal experience on any device. The interface maintains the visual integrity of the data regardless of screen size, prioritizing accessibility through adequate contrast and well-defined interactive elements.

This progress visualization constitutes a fundamental tool for the student's metacognitive reflection, allowing them to identify patterns in their learning and make informed decisions about their future linguistic practices.

6.4 Project Repositories

The system has been developed following a modern client-server architecture, with clear separation between presentation and business logic. All source code is publicly available on GitHub under the MIT license, promoting transparency, reproducibility, and community collaboration.

6.4.1 Repository Structure

- **Frontend - Client:**
  - Repository: https://github.com/EmaSuriano/language-learning-client
  - Technologies: Next.js 14, TypeScript, Tailwind CSS
  - Main components:
    - Chat interface based on Assistant UI with optimizations for learning
    - Situation and objectives selector with adaptive recommendations
    - State management with Zustand for efficient context handling
    - Internationalization system with i18n (support for 8 languages)
    - Optimized integration with audio API for voice processing
- **Backend - Server:**
  - Repository: https://github.com/EmaSuriano/language-learning-server
  - Technologies: FastAPI, Python 3.10, LangChain, Stable-Baselines3
  - Main components:
    - RAG system for contextual retrieval of educational resources
    - Integration with LLM (Phi-4) for natural dialogue generation
    - REST API with OpenAPI documentation for client-server communication
    - Optimized implementation of Faster-Whisper and Kokoro-TTS
    - Multi-Agent System system based on LangChain for agent orchestration
    - PPO model implemented with Stable-Baselines3 for level adaptation

6.4.2 Documentation

Both repositories include comprehensive documentation to facilitate understanding, use, and extension of the system:

- **General Documentation:**

  - Main README with project overview
  - Detailed installation guides for development and production environments
  - Architecture diagrams and data flow
  - Contribution guides for external collaborators

- **Technical Documentation:**
  - Complete API specification through OpenAPI/Swagger
  - Documentation of components and their responsibilities
  - Required environment variables with examples and recommended configurations
  - Guides for solving common problems

- **Examples and Tutorials:**
  - Usage examples for each main component
  - Step-by-step tutorials for custom implementations
  - Guides for extending existing functionalities
  - Examples of integration with external systems

This comprehensive documentation facilitates not only the reproducibility of the presented results but also the adaptation and extension of the system for different educational and linguistic contexts.

6.5 Current Limitations and Future Work

Despite the promising results obtained, it is important to recognize the current limitations of the system and define lines of future work to address them.

6.5.1 Identified Limitations

- **Technical Limitations:**
  - Insufficient accuracy of the STT system for strong non-native accents
  - Relatively high initial system load time (5-8 seconds)
  - Dependency on internet connection for advanced functionalities
  - Significant computational resource consumption for low-end devices

- **Pedagogical Limitations:**
  - Limited coverage of domain-specific situations (technical, legal, medical)
  - Evaluation system not yet validated with formal educational methodologies
  - Insufficient adaptation to different learning styles

  - Absence of mechanisms for collaborative learning among students

- **Validation Limitations:**
  - Reduced sample in preliminary tests (n=10)
  - Relatively short evaluation period (2 weeks)
  - Absence of control group for comparison with traditional methods
  - Lack of longitudinal evaluation of the impact on learning

6.5.2 Future Work

Based on the identified limitations and preliminary results, the following lines of future work are proposed:

- **Comprehensive Evaluation:**
  - Study design with expanded sample (n>100) and geographic diversification
  - Implementation of longitudinal evaluation (3-6 months) to measure real impact
  - Development of comparative methodology with control group using traditional methods
  - Validation with language teaching professionals and pedagogy experts
- **Technical Improvements:**
  - Refinement of the PPO model through fine-tuning with real user data
  - Specific adaptation of the STT system to improve accuracy with non-native accents
  - Optimization of the RAG knowledge base with thematic expansion and automatic updating
  - Implementation of offline modes for basic functionalities without connection
- **Functionality Expansion:**
  - Development of specific modules for specialized domains (business, tourism, academia)
  - Implementation of social component for collaborative practice among students
  - Integration with authentic content (news, videos, podcasts) for contextual immersion
  - Development of adaptive gamification system to increase motivation and retention

These lines of future work represent a structured plan to address current limitations and expand the capabilities of the system, with the aim of maximizing its educational impact and improving the learning experience for a wider range of users.

The results presented in this chapter, although preliminary, provide initial evidence on the viability and potential of the proposed approach. In the next chapter, the general conclusions

of the work will be discussed, analyzing the contributions made, the lessons learned during development, and the broader implications of these advances for the future of technology-assisted language learning.

Conclusions

7

This chapter presents the conclusions derived from the development and implementation of the language learning system based on RL techniques and Transformers architectures. It analyzes the achievements, contributions made, limitations identified, and future lines of research and development.

7.1 Project Achievements

The present work has succeeded in developing a comprehensive language learning system that meets the initially set objectives:

- **Adaptive personalization**: A system based on PPO has been successfully implemented that optimizes the learning path according to the individual profile and progress of the student, dynamically adjusting the difficulty of the content.
- **Advanced conversational interaction**: The integration of LLM models and a RAG system has enabled the generation of contextualized and natural dialogues, providing a realistic conversational experience.
- **Development of comprehensive language skills**: The system successfully integrates voice processing technologies (TTS and STT) for the simultaneous development of listening comprehension and oral production skills.
- **Efficient knowledge management**: The implementation of the RAG system provides contextualized access to relevant educational resources, improving the accuracy and relevance of the system's responses.
- **Modular and extensible architecture**: The system design allows for its evolution and adaptation to new requirements, facilitating the incorporation of improvements and new functionalities.

7.2 Contributions

The main contributions of this work to the field of AI-assisted language learning are:

7.2.1 Technical Advances

- **PPO model optimized for education**: A PPO model specifically adapted to the educational context has been developed, capable of making informed pedagogical decisions based on multiple performance metrics.
- **Effective integration of LLM and RAG**: The system demonstrates an efficient implementation of the combination of large language models with retrieval-augmented generation, providing contextually relevant and educationally significant responses.
- **Optimized voice processing pipeline**: The adaptation of Faster-Whisper and Kokoro-TTS for the educational context represents a significant optimization in terms of efficiency and accuracy for language learning applications.

7.2.2 Methodological Contributions

- **Multidimensional evaluation framework**: A systematic approach has been developed to evaluate both the technical performance of the system and its real educational impact.
- **Generation of representative scenarios**: The methodology developed for creating and evaluating representative learning scenarios provides a useful framework for future research in adaptive systems.
- **Student-centered design**: The project has implemented an approach that prioritizes the student experience, adapting technologies to real pedagogical needs.

7.3 Limitations of the Work

Despite the achievements, it is important to recognize the current limitations of the system:

- **Preliminary evaluation**: The tests conducted, although promising, have been limited to a small group of users in a controlled environment. More extensive and longitudinal studies are required to fully validate the effectiveness of the system.
- **Linguistic coverage**: Although the system supports multiple languages, the quality and depth of the educational resources vary significantly among them, with greater robustness in major languages such as English and Spanish.
- **Computational resource dependency**: The current system requires considerable computational resources, which may limit its accessibility in environments with technological constraints.
- **Cultural aspects of language**: The system shows limitations in understanding and generating culturally specific aspects of language, such as idioms, humor, or local cultural references.

7.4 Future Lines

This work opens various lines of research and future development:

7.4.1 Short-term Technical Improvements

- **Expansion of the knowledge base**: Expand and enrich the knowledge base of the RAG system, incorporating more diverse and updated educational resources.
- **Improvement of the correction system**: Implement more sophisticated techniques for the detection and correction of linguistic errors in real-time.

7.4.2 Long-term Vision

- **Multimodal systems**: Integrate multimodal understanding and generation (text, voice, gestures, facial expressions) for a more immersive and complete learning experience.
- **Adaptation to specific contexts**: Develop specialized versions of the system for specific educational contexts, such as language teaching for specific purposes (tourism, business, medicine, etc.).
- **Collaborative learning**: Explore the integration of collaborative learning systems that encourage interaction between students and the co-creation of knowledge.

7.5 Final Reflections

The development of this system represents a significant step toward the effective personalization of language learning through AI technologies. The combination of RL, Transformers architectures, and RAG systems demonstrates the potential of current technologies to fundamentally transform the educational field.

The main value of the system does not lie solely in its technical capabilities, but in its potential to democratize access to personalized and effective learning experiences. Dynamic adaptation to individual needs allows for overcoming the limitations of traditional approaches, which often fail to provide the specific support that each student requires.

However, it is important to recognize that technology, no matter how advanced, represents only a tool at the service of broader pedagogical objectives. The developed system does not aim to replace human educators, but to complement their work, providing an environment for constant practice and feedback that enriches the overall educational experience.

The true measure of this project's success will be its ability to facilitate language learning in a more efficient, inclusive, and motivating way, thus helping to break down the linguistic barriers that separate people and communities in an increasingly interconnected world.

As a final reflection, it is worth highlighting that the field of AI applied to education is constantly evolving, and this work represents only a starting point for future developments that will continue to transform the way we learn and teach languages.

Bibliography

Abdin, M., Aneja, J., Behl, H., Bubeck, S., Eldan, R., Gunasekar, S., Harrison, M., Hewett, R. J., Javaheripi, M., Kauffmann, P., Lee, J. R., Lee, Y. T., Li, Y., Liu, W., Mendes, C. C. T., Nguyen, A., Price, E., de Rosa, G., Saarikivi, O., Salim, A., Shah, S., Wang, X., Ward, R., Wu, Y., Yu, D., Zhang, C., y Zhang, Y. (2024). Phi-4 technical report.

Anderson, J. R. y Boyle, C. F. (2020). Adaptive learning systems in modern education. *Journal of Computer Assisted Learning*.

Baker, R. S. y Inventado, P. S. (2014). Educational data mining and learning analytics. *Learning Analytics*.

Brown, T. et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*.

Coyle, D., Hood, P., y Marsh, D. (2010). *CLIL: Content and language integrated learning*. Cambridge University Press.

Ellis, R. (1994). The study of second language acquisition. *Oxford University Press*.

Fries, C. C. (1945). *Teaching and learning English as a foreign language*. University of Michigan Press.

Gouin, F. (1892). *The art of teaching and studying languages*. Heath, D.C.

Graves, A., Mohamed, A.-r., y Hinton, G. (2013). Speech recognition with deep recurrent neural networks. *IEEE International Conference on Acoustics, Speech and Signal Processing*.

Hexgrad (2025). Kokoro-82m (revision d8b4fc7).

Hymes, D. (1972). *On communicative competence*. University of Pennsylvania Press.

Krashen, S. D. (1982). *Principles and practice in second language acquisition*. Pergamon.

Lewis, P. et al. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*.

Liu, J. et al. (2023). Modelos de lenguaje son aprendices de pocas muestras. *Advances in neural information processing systems*.

Nunan, D. (1989). *Designing tasks for the communicative classroom*. Cambridge University Press.

Nussbaum, Z., Morris, J. X., Duderstadt, B., y Mulyar, A. (2024). Nomic embed: Training a reproducible long context text embedder.

Peng, Y., Tian, J., Chen, W., Arora, S., Yan, B., Sudo, Y., Shakeel, M., Choi, K., Shi, J., Chang, X., weon Jung, J., y Watanabe, S. (2024). Owsm v3.1: Better and faster open whisper-style speech models based on e-branchformer.

Richards, J. C. y Rodgers, T. S. (2000). *Approaches and methods in language teaching*. Cambridge University Press.

Rodríguez, J. et al. (2023). Aprendizaje de lenguas con modelos de lenguaje de gran escala: desafíos y oportunidades. *Computer Assisted Language Learning*.

Roll, I. y Wylie, R. (2018). Learning analytics and ai: Politics, pedagogy and practices. *British Journal of Educational Technology*.

Schulman, J. et al. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Taylor, P. (2009). Text-to-speech synthesis. *Cambridge university press*.

VanLehn, K. (2011). The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*.

Vaswani, A. et al. (2017). Attention is all you need. *Advances in neural information processing systems*.

Williams, R. y Chen, D. (2017). The use of reinforcement learning algorithms in adaptive education. *Journal of Educational AI*.

Appendix: Faster Whisper and Transcription Models

A

This appendix explores Faster Whisper, an optimized implementation of OpenAI's Whisper model for speech-to-text transcription and translation. It analyzes its main features, architecture, and compares performance among the different available models.

A.1 Main Features

Faster Whisper represents a significant improvement over the original Whisper implementation, standing out for:

- **CTranslate2 Optimization**: Uses the CTranslate2 toolkit to optimize model inference.
- **Lower Memory Consumption**: Significantly reduces memory usage through quantization techniques.
- **Hardware Acceleration**: Efficiently leverages CPU and GPU through parallelization.
- **Voice Detection**: Integrates VAD (Voice Activity Detection) to improve accuracy.

A.2 System Architecture

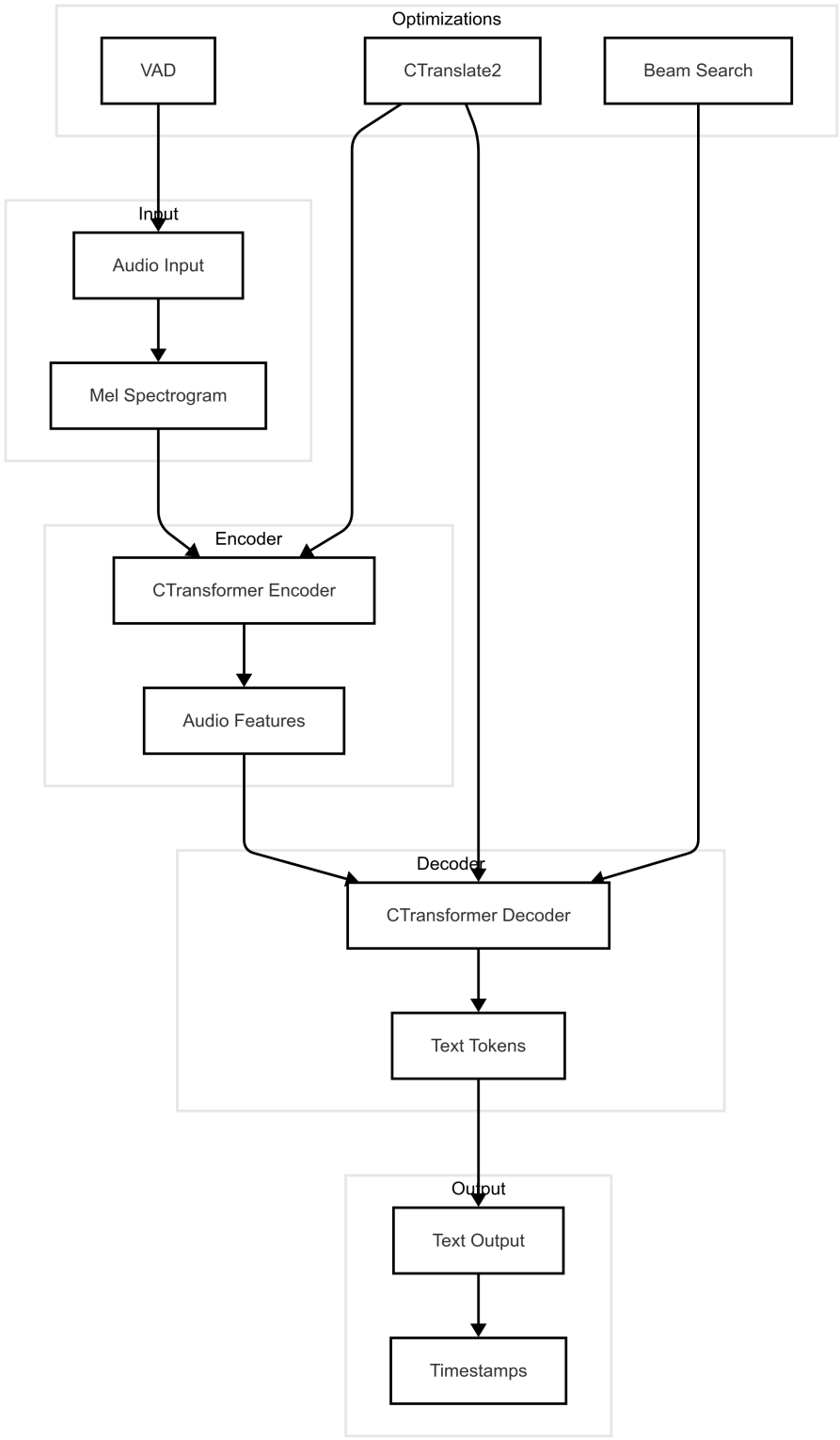

Figure A.1: *Faster Whisper Architecture*

The Faster Whisper architecture consists of several specialized modules working together to provide efficient transcription:

A.2.1 Main Components

A.2.1.1 Audio Preprocessing

The system processes audio input through:

$$\text{mel} = \log(\max(\text{STFT}(x), \epsilon)) \tag{A.1}$$

where STFT is the Short-Time Fourier Transform and ϵ is a small value for numerical stability.

A.2.1.2 CTransformer

The implementation uses CTranslate2 to optimize:

- **Encoder**: Processes the mel spectrogram into audio representations.
- **Decoder**: Generates text tokens through cross-attention.

A.3 Whisper Models Comparison

Model	Parameters	RAM (FP16)	WER	Relative Speed
Tiny	39M	1GB	7.1%	32x
Base	74M	1.5GB	6.1%	16x
Small	244M	2.5GB	5.2%	8x
Medium	769M	4.5GB	4.3%	4x
Large	1550M	7.5GB	3.6%	1x

Table A.1: *Whisper models comparison*

A.3.1 Features by Model

- **Tiny**:
  - Ideal for devices with limited resources
  - Best option for real-time transcription
  - Acceptable performance in clean audio conditions
- **Base**:
  - Balance between performance and resources
  - Suitable for web applications
  - Good performance across multiple languages

- **Small**:
  - Significant accuracy improvement over Base
  - Robust support for multiple accents
  - Reliable detection of language changes
- **Medium**:
  - High accuracy in challenging conditions
  - Excellent performance with noisy audio
  - Advanced punctuation capabilities
- **Large**:
  - Maximum available accuracy
  - Best performance on complex audio
  - Superior translation capability

A.4 Optimizations

A.4.1 Quantization Techniques

Faster Whisper implements several quantization techniques:

- **INT8**: Reduces model size by 4x with minimal accuracy loss
- **INT16**: Balance between accuracy and size
- **FLOAT16**: Maximum accuracy with reduced memory

A.4.2 Parallelization

The system implements multiple levels of parallelization:

- **Batch Processing**: Processes multiple segments simultaneously
- **Thread Pooling**: Optimizes CPU utilization
- **GPU Acceleration**: Leverages CUDA for parallel processing

A.5 Implementation Considerations

A.5.1 Model Selection

Model choice should consider:

- **Available Resources**: Memory and processing capacity
- **Latency Requirements**: Necessary response time
- **Required Accuracy**: Error tolerance

A.5.2 Deployment Strategies

Deployment considerations:

- **Edge Computing**: On-device processing for lower latency
- **Server-Side**: Greater processing capacity but higher latency
- **Hybrid**: Combination according to specific needs

Appendix: Kokoro TTS

B

This appendix explores Kokoro TTS, an open-source speech synthesis model that stands out for its efficiency and quality comparable to larger models, despite having only 82 million parameters. The model implements a lightweight architecture based on StyleTTS 2 and ISTFTNet, designed to deliver high-quality speech synthesis with limited computational resources.

B.1 System Architecture

The Kokoro TTS architecture is based on two main components: StyleTTS 2 and ISTFTNet. This combination enables efficient speech synthesis while maintaining high output quality.

B.1.1 Main Components

- **Misaki G2P**: Grapheme-to-phoneme conversion system that supports multiple languages.
- **Style Encoder**: Encodes voice style characteristics from reference audio.
- **Decoder**: Generates acoustic features based on phonemes and style.
- **ISTFT Network**: Performs final audio synthesis through inverse Fourier transform.

B.2 Technical Features

B.2.1 Model Specifications

- **Parameters**: 82 million
- **Base Architecture**: StyleTTS 2 + ISTFTNet
- **License**: Apache 2.0
- **Audio Format**: 24kHz, mono

B.2.2 Dataset

Training was conducted exclusively with permissible audio data:

- Public domain audio

- Audio with permissive licenses (Apache, MIT)
- Synthetic audio from commercial models

B.3 Voice Analysis

B.3.1 Grading System

The system evaluates voices using two main metrics:

- **Objective Quality**:
  - A: Exceptional quality
  - B: Good quality
  - C: Acceptable quality
  - D: Limited quality
- **Training Duration**:
  - HH: 10-100 hours
  - H: 1-10 hours
  - MM: 10-100 minutes
  - M: 1-10 minutes

B.3.2 Voice Distribution

Language	F	M	Total	Average Quality
American English	11	9	20	B-
British English	4	4	8	C+
Japanese	4	1	5	C+
Mandarin Chinese	4	4	8	D+
Spanish	1	2	3	C
French	1	0	1	B-
Hindi	2	2	4	C
Italian	1	1	2	C
Brazilian Portuguese	1	2	3	C

Table B.1: *Voice distribution and quality by language*

B.4 Performance and Limitations

B.4.1 Optimal Operating Ranges

Model performance varies according to text length:

- **Optimal Range**: 100-200 tokens
- **Reduced Performance**: <20 tokens
- **Possible Acceleration**: >400 tokens

B.4.2 Training Costs

Kokoro's training has been remarkably efficient:

- **GPU Hours**: 1000 hours on A100 80GB
- **Total Cost**: Approximately $1000 USD
- **Average Rate**: $1/hour

B.5 Comparison with Other Models

Model	Parameters	Voices	Languages	License
Kokoro	82M	54	8	Apache
Coqui	1000M	1087	100+	MIT
Bark	900M	100+	100+	MIT

Table B.2: *Comparison with similar TTS models*