LAB University of Applied Sciences

Interactive Language Learning Application using Artificial Intelligence

LAB University of Applied Sciences
Bachelor’s degree in information and communication technology (ICT)
2024
Jordi Linares Carrasquer

Abstract

Author(s)	Publication type	Completion year
Linares Carrasquer, Jordi	Thesis, UAS	2024
	Number of pages	
	57	

Title of the thesis

Interactive Language Learning Application using Artificial Intelligence

Subheading, if any

Degree, Field of Study

Information and Communication Technology

Name, title and organisation of the client

Ismo Jakonen, Senior Lecturer, Media Technology

Abstract

This thesis explores the convergence of artificial intelligence (AI) and language learning through the development of a mobile application. The application employs AI-driven image captioning to recommend expressions and phrases based on real-world scenes captured by users' cameras. Emphasizing simplicity and interactivity, the app enables users to listen to correct pronunciation, engage in pronunciation tests, and store learned phrases for future review. The study delves into the theoretical foundations of AI, language processing, and development methodologies. The practical implementation results in a user-centric language learning tool that redefines traditional approaches, offering an accessible and engaging platform for immersive language learning.

Keywords

Artificial Intelligence, Education, Language Learning, Natural Language Processing, API, Mobile Application, Image Captioning, Speech Recognition, Machine Learning, Deep Learning, Interactive

Contents

List of abbreviations/concepts/terms

AI: artificial intelligence

API: application programming interface

Framework: a tool that provides ready-made components or customised solutions to speed up development.

LLM: Large Language Model

LMM: Large Multimodal Model

NLP: Natural Language Processing

1 Introduction

In an era dominated by technological advancements, the fusion of artificial intelligence (AI) and mobile applications has opened unprecedented avenues for innovative learning experiences. This thesis embarks on a journey to harness the power of AI in the realm of language learning through the creation of a cutting-edge mobile application. Unlike conventional language learning apps, this approach prioritizes simplicity and interactivity, allowing users to integrate language practice into their daily lives.

This thesis is driven by the goal of exploring the potential of AI in language education, with a specific focus on practical and interactive applications. The theoretical foundation will delve into the evolution of AI, emphasizing its role in language processing, machine learning, and deep learning. A comprehensive understanding of frontend and backend development will be presented, ensuring a grasp of the technical facets involved in creating an engaging and user-friendly language learning interface.

The practical facet of this study converges on the development of a mobile application that not only facilitates language learning but also seamlessly integrates AI capabilities into the user experience. The application's user-friendly design ensures accessibility and ease of use, aligning with the modern lifestyle where learning can happen anywhere, anytime. Through the implementation of image captioning models and interactive language exercises, this application aspires to redefine language learning paradigms, making it both engaging and effective, ensuring the elimination of the feeling of repetitiveness in conventional language learning applications.

The objective is not merely to create an innovative language learning application but to contribute to the broader discourse on the transformative potential of AI in education. By pushing the boundaries of conventional language learning methodologies, this work seeks to empower users with an interactive, AI-driven tool that transcends linguistic barriers and achieves a dynamic and personalized learning experience.

The heart of this application lies in its unique feature that gets the maximum of AI-driven image captioning and speech recognition. Users can effortlessly capture real-world scenes using their mobile cameras, initiating a dynamic process that unfolds in three key parts. First, the image undergoes analysis, generating recommended expressions and phrases pertinent to the captured context. Users can then engage in immersive learning by listening to the correct pronunciation, to then be able to perform some pronunciation practice by themselves, and finally storing the learned phrases for future review.

2 Language Learning

2.1 Importance of language learning

Language learning is multifaceted and can not only create a communication convenience but also open an entrance for such acceptance of different cultures and many ways of thinking (Arnold 2019; Bozorovich 2021). Professionally, it is more pronounced today since many professions require multilingual skills (Kioko 2018). The significance of language learning is as well enviable by its potential of helping develop identities of individuals for the individual learner, group identity for racial or ethnic communities, national identity for countries, and human possibilities for society's individuals (Warahmah 2022).

Multilingualism denotes a speaker's ability to articulate thoughts in multiple languages with proficiency like a native speaker. Multilingualism can also be understood as the coexistence of numerous languages within a society, including official or unofficial, native or foreign, and national or international languages (Okal 2014).

Proficiency in multiple languages enables effective communication with a diverse range of individuals in both personal and professional settings. The cultivation of multilingualism has been found to boost cognitive adaptability and creativity. Contemporary research suggests that children raised in a supporting environment where they engage with multiple languages from a young age tend to possess higher perception and intellectual flexibility compared to their monolingual counterparts. Numerous credible findings support the improvement of cognitive flexibility. Dr. Elsie Naude, a speech and language therapist from Pretoria, observed during her in-depth research on multilingualism that parents contribute to their child's intellectual development by encouraging the acquisition of additional languages. As a result, numerous children proficient in multiple languages excel in lateral thinking, demonstrate enhanced social adaptability, exhibit superior thinking and reasoning skills, and display greater cognitive abilities (Okal 2014).

Multilingualism offers a perspective into diverse cultures and experiences, transforming an individual into a multicultural entity, and giving a competitive advantage in the current job market. Employers seek not only academic and professional qualifications but also value proficiency in desired languages as an additional asset. Consequently, being multilingual is a positive attribute for any job seeker in today's era (Okal 2014).

2.2 Transforming language learning with AI

Artificial Intelligence (AI) can be traced back to a branch of computer science in which machinery is built and designed with a view of reproducing human intelligence for allowing it to learn, reason, perceive, draw conclusions, communicate, make judgments, and decisions exactly in the same way humans do. All of these are done through algorithms and machine learning technologies. AI broadly falls into two categories, which are: Narrow AI which is designed to execute a particular task, and General AI can do just any intellectual task. It is also characterized by a degree of cognitive capacity and autonomy. It is applied in a series of works, including one in the line of heavy industry, gaming, aviation, weather forecast, and expert systems. It can find its application in other many places as well (Morandín-Ahuerma 2022; Ramírez-Ortiz 2020).

Some of the benefits of the incorporation of AI into education are personalized learning experiences, organizational benefits, strong feedback mechanisms, and robust analysis of data (Zaman 2023; Lampou 2023; Harry 2023). Meaning that all these education users' benefits potentially generate improved students' outcomes as well as increased teachers' effectiveness (Harry 2023).

However, responsible integration of AI in education is necessary since it can bring difficulties such as privacy and security concerns, inadequacy of trust, cost, and likely bias. Despite such challenges, AI in education has the potential of offering better data analysis that helps educators to make appropriate decisions based on data (Harry 2023).

2.3 AI applications in language learning

The intrusion of AI into the domain of language education has the potential to yield radical results regarding the teaching and learning experiences of the students as well as the teachers. Also, AI-powered tools and systems have opened new doors about teaching foreign languages. A major advantage of AI in this area is the scalability of handling large datasets, making its algorithms identify common language usage tendencies, zooming on frequently occurring errors to make learners enjoy well-analyzed personalized feedback. AI can help people learning a foreign language get customized insights that fit their exact needs and wishes. Such tailor-made tutoring will make the learning journey more effective whereby the students are allowed to focus on their weak points with pin-point advice according to his or her aptitude (Chisega-Negrilă 2023).

Here are some of the user cases where AI is used in language learning applications:

1. **Personalized Learning Paths**: Utilizing machine learning algorithms, AI analyzes student data to customize learning experiences, providing resources, identifying areas for improvement, and adjusting task difficulty levels. This approach not only helps struggling students catch up but also allows advanced learners to be adequately challenged, leading to increased engagement, better academic performance, and higher retention rates. AI in education takes various forms, such as analyzing past performance for targeted support and adapting to a student's learning pace and has been applied successfully in diverse educational settings from K-12 to corporate training (Harry 2023).

2. **Interactive Chatbots:** AI chatbots can significantly enhance spoken language teaching and learning, as they engage learners in computer-simulated human-human conversations (Sha 2009). Through their conversational interface, chatbots have the potential to enhance the interactive and engaging nature of learning, thereby encouraging active participation and boosting student motivation. Additionally, chatbots can introduce gamification into the learning process by providing rewards and incentives as students complete tasks and achieve their educational objectives (Harry 2023).

3. **Automated Grading**: With the use of AI the automatization of the correction and grading processes helps teachers save effort at the same time it provides learners immediate feedback. By utilizing automated assessments, AI can assess individuals' speaking, writing, listening, and reading abilities and promptly offer feedback on areas that require enhancement (Chisega-Negrilă 2023).

4. **Progress Tracking**: AI has the capability to monitor learners' advancement throughout their learning journey and offer tailored suggestions for their ongoing language learning studies. By continuously assessing performance, AI algorithms can propose specific learning resources and activities aimed at helping learners enhance their language skills over time (Chisega-Negrilă 2023).

In summary, by offering individualized, engaging, and effective resources, AI has the potential to completely change the language learning landscape (Son 2023; Vall 2023). These resources could provide customized learning experiences, reduce the learning curve, and introduce students to various cultures. However, they face challenges like the need for more human contact and dependence on large datasets (Vall 2023).

2.4 State of art in AI for language learning

The use of AI in language learning has seen significant advancements, with a focus on personalized and interactive experiences (Son 2023). One notable development is the Teachable AI system, which allows for the learning of new language concepts from users during live interactions (Ping 2020). This trend is reflected in the most common keywords in AI-based language learning research, which include mobile-assisted learning, virtual reality, and machine learning (Jaleniauskiene 2023). AI technologies such as voice and image recognition are also being applied to facilitate language learning, particularly in the areas of spoken language judgment, composition correction, and emotion recognition (Zhu 2020).

Several applications have been developed that use artificial intelligence to facilitate language learning. Here are some of the most notable ones:

1. **Duolingo**

   Duolingo is a pioneering language learning platform launched in 2012. It stands out as a prime example of utilizing gamification in education technology, offering 40 languages through a user-friendly website and mobile application. Duolingo has successfully democratized language learning by providing free access to high-quality education, aiming to bridge social inequalities. This strategic approach, coupled with a robust business model that includes a mix of subscriptions, advertising, and in-app purchases, has propelled Duolingo to become the top-grossing education app since 2019. The platform's unique blend of technology, community engagement, and commitment to accessibility has not only attracted a vast user base but also significantly contributed to its competitive advantage in the rapidly evolving digital language learning industry (Sakalauské 2022).

   Most often, Duolingo's advanced AI approach has been acknowledged for its ability to offer a highly personalized course adapted to the individual learner (Bicknell 2023). This has been shown lately with the introduction of Duolingo Max, as seen in Figure 1. This adaptability has served very well when subjected to the typical pedagogic language courses. In most of the cases, Duolingo is put as an additional tool while working with vocabulary and grammar learning (Munday 2016). On the other hand, there has also been significant criticism pointed at its design and learning outcomes, with further research even pointing to how wide open the area of effectiveness is (Shortt 2021). Duolingo captivates modern learners in that it in-

troduces an element of fun and interactivity into this meticulous event of the language studied (Savvani 2019).

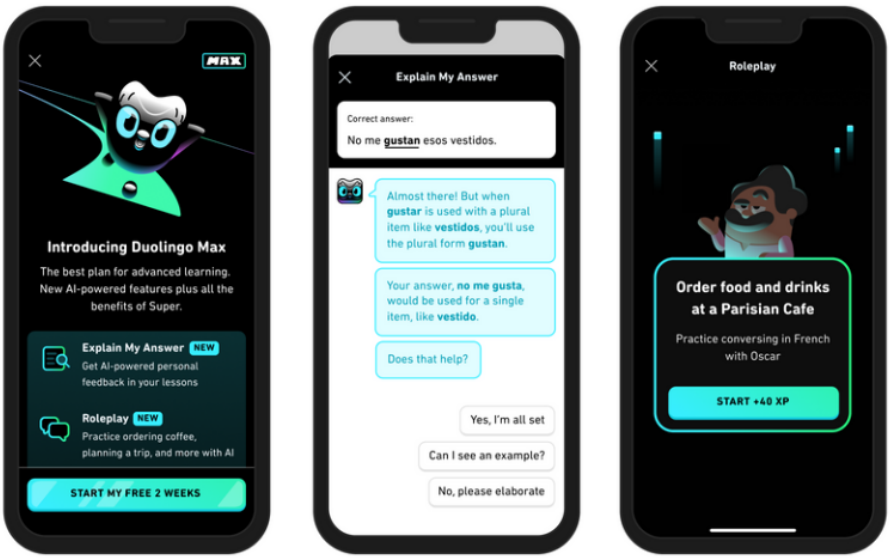

Figure 1. Duolingo Max (Duolingo Team 2023)

2. **Babbel**

Babbel provides plenty of exercises that are meant to better reading, writing, speaking, and listening skills. However, what really distinguishes Babbel from other products is the quality of the content, the sophisticated design of the platform, and the accuracy of the speech recognition technology (Emery 2022).

Babbel has launched two new AI features named "AI-Enhanced Speech Recognition" and "Everyday Conversations." The latter provides learners with custom feedback on their pronunciation skills and helps simulate standard dialogues one would experience in real life. The AI-Enhanced Speech Recognition tool, built on insights from Babbel's extensive audio library, analyzes learners' speech to give them a detailed assessment of their pronunciation, considering accents, dialects, and ranges. Meanwhile, Everyday Conversations allows users to engage in simulated dialogues on scenarios like ordering food and chatting with friends, preparing them for real-world situations, as show in Figure 2. Babbel offers these features through its app, supporting learners from absolute beginners to those well-practiced in integrating newly acquired language skills into daily life (Babbel 2023).

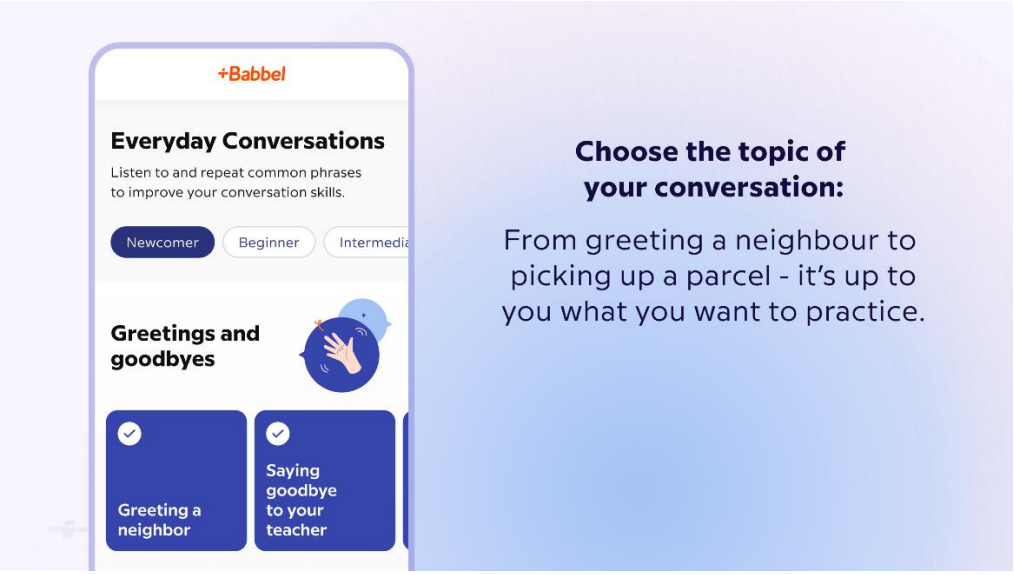

Figure 2. Babbel Everyday Conversations (Babbel 2023)

3. **Rosetta Stone**

Rosetta Stone is a long-established language learning app whose solid foundations are engaging immersion, breadth of language offerings, and live instruction to emphasize their focus on effective language learning. The platform uses AI technology to design lifelike real-world interaction simulations that plunge learners into the nuances of cultures, varied accents, and everyday language depths, offering comprehensive language learning. This provides an immersion through education that's deeper than just the words and phrases (TalkPal AI).

4. **Mondly**

Mondly is a web and mobile application platform for foreign language learning, offered to learners in a uniquely designed manner. Mondly uses a unique combination of vocabulary and phrase learning with advanced speech recognition and chatbot technologies to ensure that users are well prepared with the right skills to speak a new language. Moreover, it gives instant feedback on pronunciation and even provides help with grammar and conjugation tables—all that through a game-based practice (Mondly).

Mondly has an AI-based assistant by the name of Luna, which uses a script of dialogues generated by AI that engages the users in real conversations between two native speakers. AI can process the language of the user, his intentions, and how

the conversation is going on the fly, giving answers fitting the dialogue at that point. Figure 3 illustrates an example of this process (Mondly).

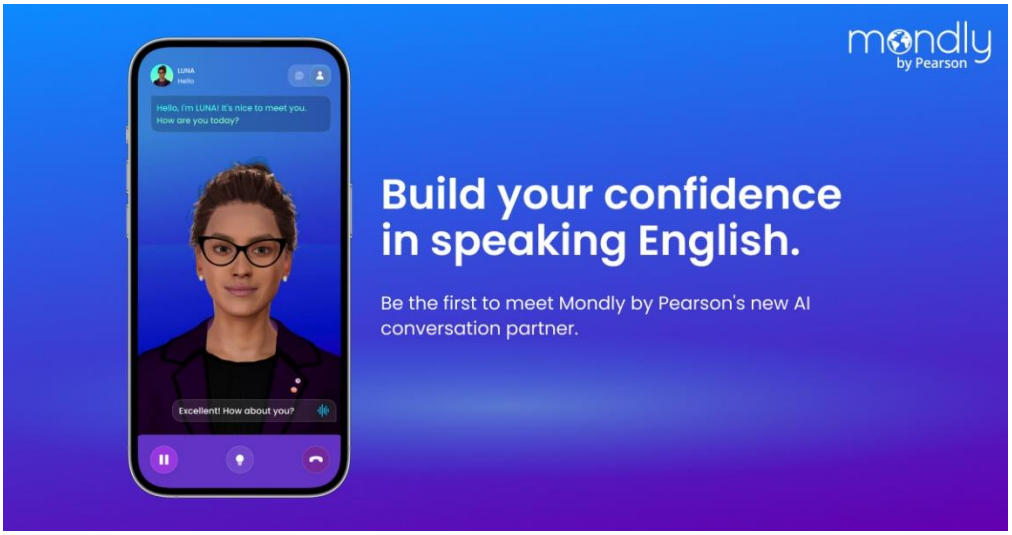

Figure 3. Luna AI assistant (Mondly)

2.5 Interaction: The key to make language learning fun

Interaction refers to the mutual influence that may be given between two or more entities, phenomena, or elements. The idea has its origin in reflection on diverse disciplines, which could consider this theory to adapt it according to requirements, keeping in mind the essential aspects. Modern sociology, anthropology, and communication, to name but a few, with the newly established area of research, human-computer interaction, have applied its aspects (Frozzini 2021).

Interaction is a critical component in learning a second language as it provides an opportunity for the language learners to engage in communicative practice and meaning-making activities (Gąsior 2019). In the case of adult learning, when interaction between people and interaction with the context is claimed to be crucial, this tendency is even more vivid. (Hansman 2001). In autonomous language learning, interactive activities serve as an important stimulation for the development of students' autonomy as well as communicative competence. Interactive activities empower students to become active participants in their learning journey, fostering an environment conducive to creativity and productive engagement in education. Engaging in these activities cultivates attributes essential for active learning and effective foreign language usage, including the ability to make independent choices and collaborate in communication. These attributes lay the groundwork for achieving autonomy in language learning (Putistina 2015).

Effective teaching of foreign languages includes teaching within a continuum of interactive circumstances. Such an approach realizes the importance of learning a foreign language implying at least the acquisition and mastering of social competencies, knowledge of additional content, and linguistic proficiency. Attainment of this goal will be through diversification of activities in the classroom and the inherent roles both to the students and the teachers. This approach is likely to succeed in making learners grow up not only as competent users of a foreign language but also as cultured and sophisticated members of society (Gąsior 2019).

The language acquisition process goes further from simply acquiring a new linguistic structure; rather, it is an active, exploratory journey that underlines the need for social engagement and guidance toward knowledge building. It is the development of learners' ability to self-regulate through participation in mental and physical activities where their action is guided by someone else at the initial stage, and not just mere absorption of forms. This view is supported by research that social interaction is very critical to the acquisition of language learning, mainly because it accords learners inputs that are relevant to learners' proficiency level. Social interaction is one of the aspects that have the potential to deliver input to the learner that is at his proficiency level (Lin 2017).

When making a comparison between online and face to face interactions, research consistently shows that the dynamics of interactions in online and in-person settings are markedly different, and the result is also different. Online engagement often promotes an even footing among participants and focuses more on larger concerns related to writing. In-person engagement is often more hierarchical and tends to focus on grammar, vocabulary, and stylistic details (Jones, 2006). The in-person interaction brings about more favorable impressions, and it does compare with greater agreement between how one sees himself and how he is perceived by the other person than in online interaction (Okdie, 2011). As to learning and growth, synchronous classroom face-to-face interactions showed increased academic performance and elicited high levels of satisfaction and superior perceptions of the learning environment. On the contrary, various studies have characterized asynchronous online interactions as encouraging students to be independent and even facilitating the formation of full theses (Yin, 2021). With this considered, in-person learning, however, is generally regarded with more favorability in reference to social presence, interaction, and overall satisfaction (Bali, 2019).

3 Artificial Intelligence

3.1 Introduction

Artificial Intelligence (AI) is the field of computer science that aims to give machines the ability to perform tasks that typically require human intelligence, such as perception, reasoning, problem-solving, and planning (Wu, 1986). However, there are two concepts that must be defined first: intelligence and creativity.

The concept of intelligence is quite slippery, and up to now, not one universally acceptable definition has been offered to include all that it is and encompasses. Few have advanced, other than academic or professional disciplines, their versions of what intelligence entails, hence presenting a rich canvas of perspectives. This debate has been particularly salient within the domains of AI and cognitive science, as scholars and practitioners continue to struggle with identifying the core attributes that should constitute intelligence. Common threads in these discussions include cognitive capabilities such as problem-solving, perception, reasoning autonomously, planning, adaptability, and learning. Beyond these, other theories bring in the idea of intelligence to mean emotional and social intelligence. Suggesting a bigger, more tolerant understanding, as in, this view shows that there is more in intelligence than reasoning, and it shows that there is more in it than problem-solving. Moreover, the question whether consciousness is a prerequisite for intelligence further complicates the argument by implying multifaceted intelligence and the difficulty of crystallizing the gist of it (Linares et al. 2023).

Creativity, in a way, is even more defiant of a single definition than intelligence. Generally, it is considered one of the capabilities to produce ideas, products, or solutions that are novel and valuable. Creativity is the hallmark of human innovation. There is an ongoing and divided debate if computers can really produce creative ideas. On the other hand, the skeptics argue that even if computers can simulate the creative processes through algorithms and data analysis, which eventually result in new ideas, this simulation of creativity would not possess the depth that human creativity has, with considerations of emotion, intuition, and consciousness. Proponents of computational creativity, on the other hand, believe that computers will or can be creative, in the sense that they should or can bring forth novel and valuable ideas, ones that have not been previously imagined. This debate also involves the nature of creativity itself, not only the questions of its nature but also challenges the understanding of the boundaries between human and machine abilities (Linares et al. 2023).

AI is a full cross-cutting field, where the final objective is the automation of all human activities at present requiring human intelligence (Krishna 2018). Figure 4 shows exactly how AI presents itself as a multidisciplinary field. Such areas, like machine and deep learning, are reshaping society and driving advancements in data science (Górriz 2020).

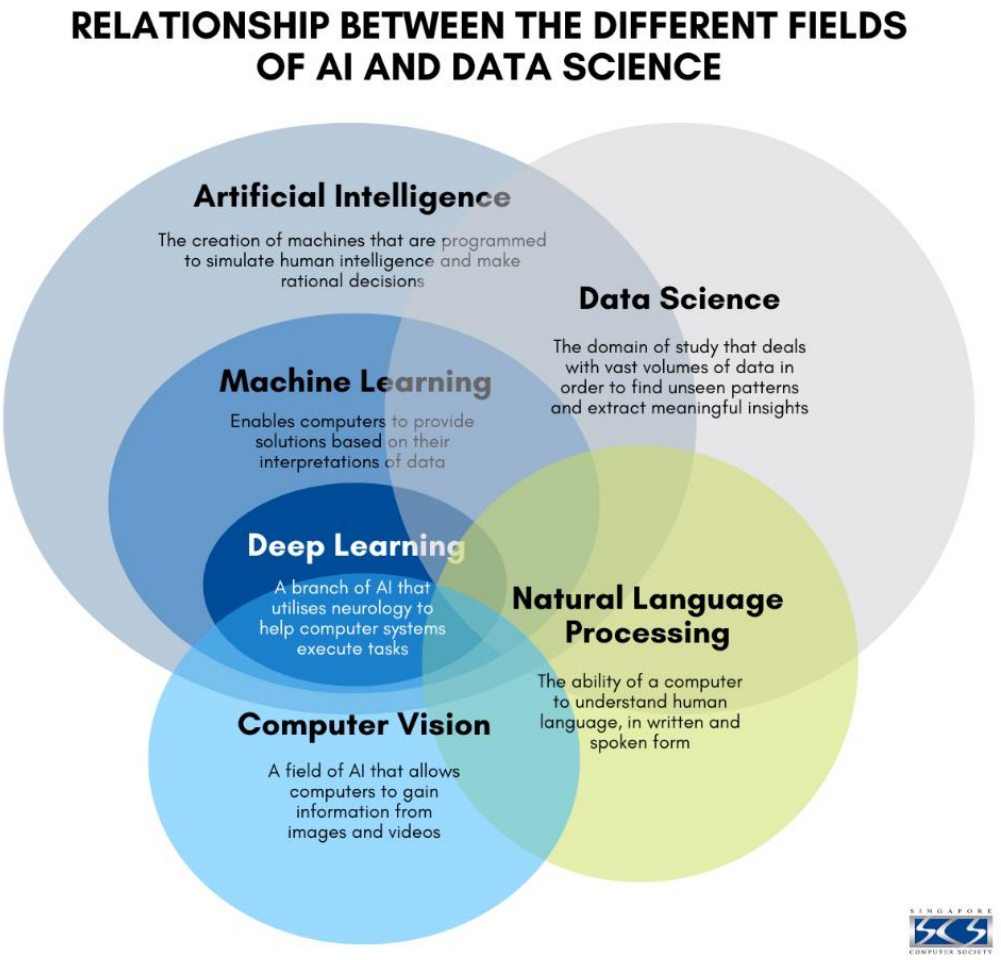

Figure 4. Main Fields of AI (Singapore Computer Society)

3.2 History

The history of AI is one that deeply marries the human imagination, philosophy, and the advancements of technologies. The notion of creating intelligent machines is one that has been around for centuries. Man dreamed up creating intelligent mechanisms from the very beginning, inspired by the ancient descriptions of texts telling fictional accounts of mechanical beings in the service of appearing gods. It was only later, after some time, that philosophers including René Descartes and Gottfried Wilhelm Leibniz considered the idea of mechanical reasoning and the possibility of the logical operations of a machine, hence paving way for later research in the field of AI. Such early philosophical speculations developed into actual experiments with calculating machines in the 17th and 18th centuries. These very definite achievements demonstrated the possibility of mechanization in arithmetic operations (Buchanan 2005).

Development in technology witnessed during the 20th century was immensely based on electronics and computing, further pushing AI from theoretical talks to practical experiments. The work by Alan Turing proposing a human-like intelligent machine was undertaken during the mid-20th century. It was at this time that these early computers, called "giant brains" because of their vast potential to perform difficult calculations and work at paces unimaginable before, came onto the scene. Turing's work led to the development of the Turing Test, a method for determining whether a machine can demonstrate human-like intelligence, which is shown in Figure 5. Turing's ideas laid the groundwork for the development of a programming language and computational device that could test hypotheses about intelligence and behaviour (Buchanan 2005).

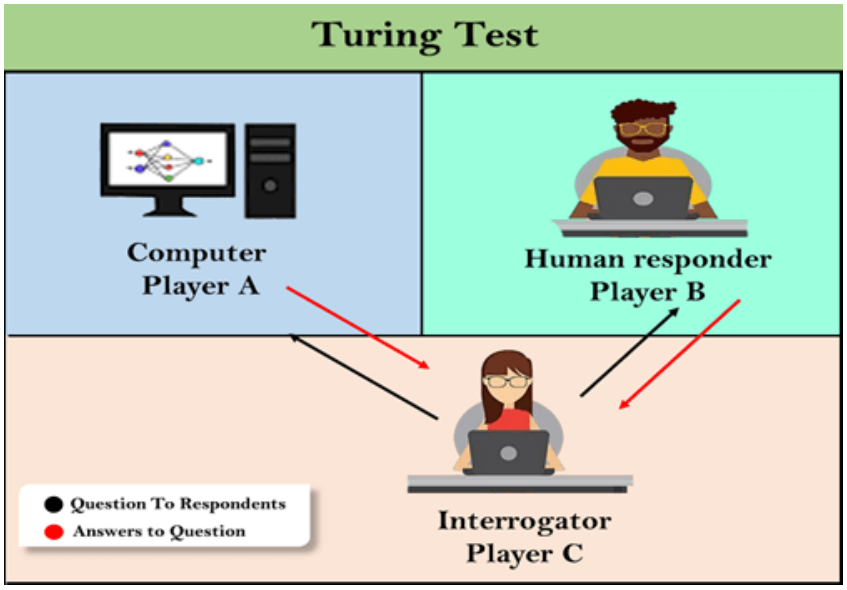

Figure 5. Turing Test (JavaTPoint a.)

In the summer of 1956, the Dartmouth College gathering marked the inception of AI as an academic discipline. This pivotal two-month conference, founded on the hypothesis that all aspects of learning and intelligence could be so precisely defined that a machine could replicate them, was where the term "Artificial Intelligence" was coined. It also served as the initial meeting point for future AI pioneers such as John McCarthy, Marvin Minsky, Allen Newell, and Herbert Simon. Due to its significance in bringing together these leading figures and introducing the term AI, the Dartmouth Conference is regarded as the cornerstone of the formal study of AI (Klassner 1996).

The period from the 1960s through the 1970s was characterized by a large amount of funding from the Department of Defense to AI research, and this resulted in the development of systems that could emulate human problem-solving and natural language understanding (Russell 2015). This progress was followed by a disillusioned period in the late 1970s and 1980s, known as the 'AI winter,' due to the narrow capabilities of the technology (Shin 2019).

A key turning point that set such a change and progress in motion within the domain of AI was the renewal of AI in the '90s, mainly driven by machine-learning approaches (Jayabharathi 2021). It was a time of shifting from rule-based systems to algorithms that learned from and based their decisions on data. This resurgence has been in part thanks to the role microprocessor architecture has played, besides increased capabilities of microprocessors that have enabled new implementations of complex, intelligent, real-time applications (Khan 2021). Deep Blue, a chess-playing computer, made history in 1997 by defeating the reigning world chess champion, Garry Kasparov, in a six-game match, depicted in Figure 6. This victory was a significant milestone in the field of AI, demonstrating the potential for progress in related AI problems (Newborn 2000).

Figure 6. Garry Kasparov vs Deep Blue (Goodrich 2021)

2012 brought a major transformation with the arrival of AlexNet, an advanced deep learning algorithm that made many breakthroughs in image recognition. This breakthrough during the ImageNet Large Scale Visual Recognition Challenge really proved the giant potential of deep convolutional neural networks (CNNs) by reducing error rates much greater than anything else available before. The success of AlexNet was attributed to the very deep CNN architecture, and it developed a new way of activation using rectified linear units (ReLU) and innovative training techniques such as dropout regularization, one of the many key components required for efficient training of large NNs. This paradigm shift has brought to light the capabilities of deep neural networks in dealing with complex data patterns and catalysed tremendous technological advancements and applications witnessed in industries from healthcare to autonomous vehicles, in which machine learning and data analysis pervaded (Kamali 2023).

Modern AI is a rapidly evolving field, encompassing techniques such as machine learning, deep learning, natural language processing, and robotics. The development in AI technologies has reached amazing levels of precision in such areas as image and speech recognition, natural language understanding, and game playing. The success is largely attributed to deep learning, where large artificial neural networks are trained on large datasets to look for patterns that enable extremely precise prediction. On top of that, modern AI has seen overwhelming progress in generative models applied to reinforcement learning. However, even with these milestones, the advancement of AI journey faces a few hurdles which include—in issues around data privacy, imperatives for fairness and bias mitigation,

AI systems that urgently need to offer clear, explainable rationales for the decisions they make (Arunagiri 2023).

3.3 Types of AI

Classifying AI can be approached from various perspectives, depending on the criteria or dimensions considered. Here are some common ways to classify AI:

Based on Capabilities

- Specific or Narrow AI: What exists today under the label of Artificial Narrow Intelligence (ANI), or Weak AI for short, is ANI. Every other classification of AI is applied only on paper, not in practice. ANI can do something specific, and that's all: a specific, narrow task better, faster, and with fewer errors than a human. However, its functionality is limited to what it has been programmed to do. It is designed to focus on advancing within a given domain of cognitive skills. Examples of Narrow AI include Siri by Apple, Amazon's Alexa, and IBM's Watson. OpenAI's ChatGPT will also fall in this category since it can carry out only text-based conversation. In other words, showing the limitation to only one function (IBM a. 2023).

- General or Strong AI: The way AI is being discussed here is "Artificial General Intelligence" (AGI), although it is really a theoretical concept up until now. AGI refers to the level of AI that will utilize knowledge and skills applied from past experience to autonomously meet challenges in various new contexts without having human beings retrain it. This characteristic enables AGI to undertake and master any cognitive task that humans are capable of performing (IBM a. 2023).

- Super AI: Artificial Superintelligence, often termed Super AI, continues to remain theoretical ideas similar to the other one, AGI (Artificial General Intelligence). If it really happens, Super AI would be able to reflect much higher-ordered cognitive abilities than human beings, e.g., thinking, reasoning, learning, making judgements. Powered by Super AI, the applications go beyond the understanding of human emotions and experiences to actually having the ability to experience feelings, have needs, and hold beliefs and desires by themselves (IBM a. 2023).

Figure 7 shows in detail these types of AI.

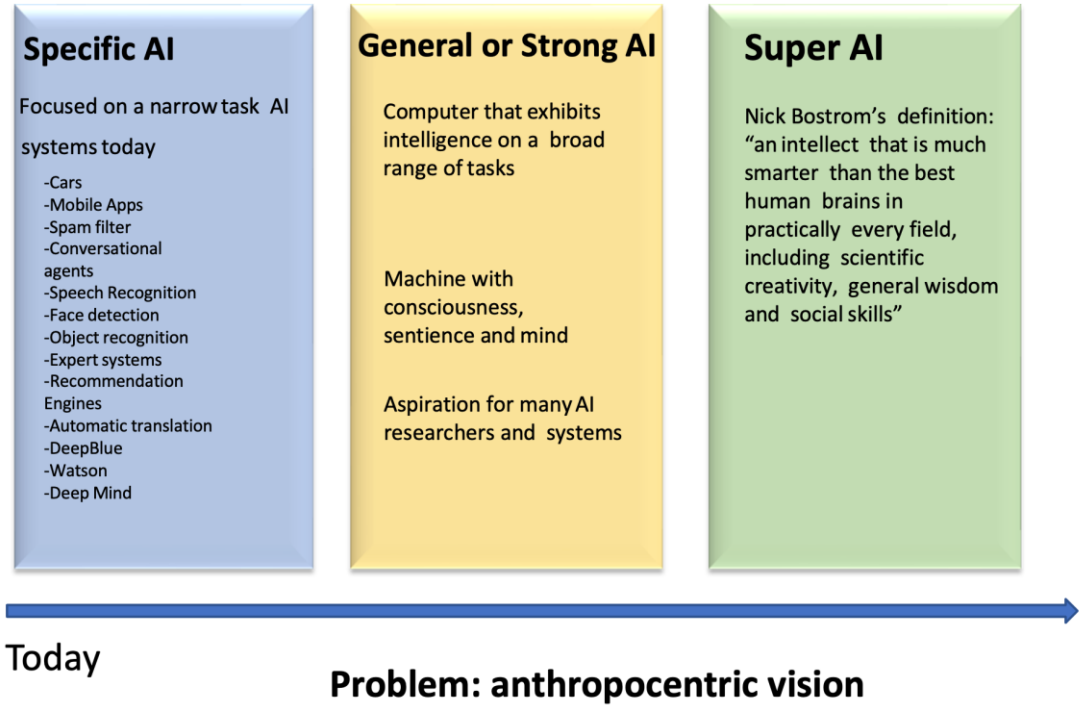

Figure 7. Types of AI by capabilities (Linares et al. 2023)

Based on Functionalities

• Reactive Machine AI: Reactive machine AI works without memory and is specialized for pretty narrow tasks. It makes use of statistical mathematics to sort through huge datasets for instantaneous data processing and simulating intelligent behavior without calling upon actions taken or outcomes produced. A prime example is Deep Blue, the IBM supercomputer, which won over grandmaster Garry Kasparov in the 1990s through analyzing current board configurations and projecting possible move outcomes. The Netflix Recommendation Engine uses algorithm analytics on historical data of viewers in order to give users personalized suggestions for content, counting on the kind of content they are most likely to prefer (IBM a. 2023).

• Limited Memory AI: Limited Memory AI has the capability to remember the past events or the outcomes of a certain duration. It can thus keep track of objects or situations over time. This gives them a chance to make informed decisions that use historical and current data, although they do not have the capability to store that data for future reference. Such AI gets better with more data over time. Example tools of Generative AI include ChatGPT, Bard, DeepAI, etc. Generative AI is used to predict content, either text or visual, and even in certain virtual assistants like Siri, Alexa, Google Assistant, Cortana, and IBM Watson Assistant, to interact and process tasks using natural language processing coupled with Limited Memory AI. Other fields

have benefited too from this technology, like the classic example of self-driving cars (IBM a. 2023).

• Theory of Mind AI: Theory of Mind AI is the advanced category of AI, much more a subset towards General AI, which can help in understanding thoughts and emotions of others for a much better and enhanced way of interaction. Although still in the stage of conception, such AI would understand human motives and emotions, so it would react differently according to individual emotional needs and intentions. This level of AI purports to be able to foster human-like relationships with personalized interactions and nuanced understandings of complex human outputs, like artwork and essays—both of which are not properly handled by current generative AI tools. Emotion AI is a branch of Theory of Mind AI currently developed and tries to interpret and respond to human emotions. However, it has yet to achieve the ability to fully understand and respond to human emotions effectively (IBM a. 2023).

• Self-Aware AI: Self-Aware AI represents an advanced, although conceptual, category of AI that would have not only superior abilities similar to super AI but also be able to understand its own internal states, feelings, needs, and beliefs as a human being does. A relevant domain would be Emotion AI, which refers to the way of using AI in sensing and interpreting the human affective state. One of the related domains of Emotion AI is Theory of Mind AI, as it strives to design a kind of AI that perceives human feelings from his or her voices, images, or any other type of data. Emotion AI, despite being in a very developed stage, has not yet concluded with the capacity to understand and react to human emotions (IBM a. 2023).

3.4 Machine Learning

Machine learning, a subset of AI and computer science, emphasizes leveraging data and algorithms to mimic human learning processes, thereby enhancing precision over time. It revolves around the creation and examination of statistical algorithms capable of learning from data, which allows computers to undertake decision-making or predictive tasks without the need for explicit programming (IBM b.).

Machine learning, as distinguished from conventional programming, is characterized by its ability to improve through experience, often using data (Kollmannsberger 2021). It is also closely related to conventional statistics, with the former being an advanced form of the latter (Dhillon 2022). Machine learning is particularly useful in tasks where there are no human experts, or where human experts are unable to explain their expertise, such as in speech recognition and natural language understanding (Dietterich 1996). It is a branch of

AI that teaches machines to learn from past experiences, and its algorithms adaptively improve as they learn from more examples (Shaveta 2023). This difference can be shown in Figure 8.

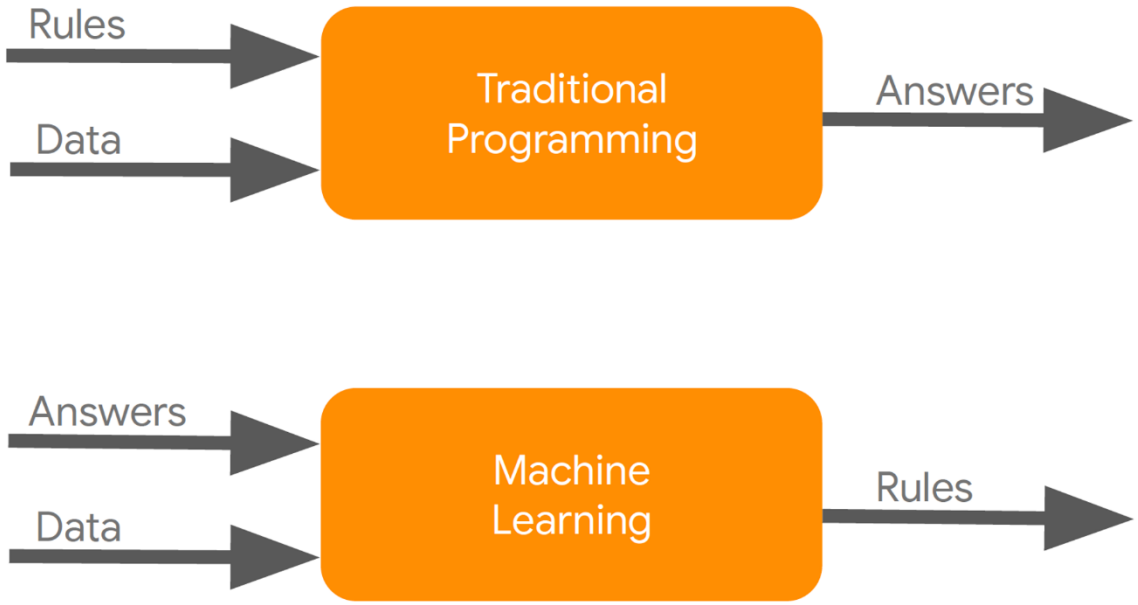

Figure 8. Traditional Programming vs Machine Learning (Linares et al. 2023)

The Training Process

Machine learning algorithms can be broadly classified into three categories depending on how the models are trained: supervised learning, unsupervised learning, and reinforcement learning.

In supervised learning, an algorithm is supplied with labeled data, such that every example to be learned from in the dataset has an associated known outcome, often called a label. This approach enables the algorithm to learn from these examples to make predictions on new, unseen data. Supervised learning can deal with two kinds of problems: classification, where the goal is to classify the data into distinct classes or groups, and regression, where the goal is to predict some continuous numerical value. These learning types are just like the guide, how some or specific outcome is taught, and the process always leans towards them. So, the model predicts or classifies well when it comes to new data based on patterns learned from training datasets (Linares et al. 2023).

Unsupervised learning, or unsupervised machine learning, utilizes machine learning algorithms to analyze and cluster input datasets that are unlabelled. Such algorithms find structure in data, detecting the subtle and often useful patterns or groupings present without human intervention. The method can look for the similarities and differences contained in the information; hence, it is the best solution for exploratory data analysis, cross-selling

strategies, customer segmentation, and image recognition. Unsupervised learning means the data is without labels; hence, hidden patterns, classes, anomalies, etc have to be figured out (Linares et al. 2023).

Reinforcement learning is a subfield of machine learning in which the agent "lives" in an environment that could sense a state at every time step as a vector of characteristics. The agent has an opportunity to take actions in each possible state. Different actions have different rewards and could also lead to the machine being in another state of the environment. The goal of a reinforcement learning algorithm is to learn a policy. A policy is a function (like the model in supervised learning) that takes the feature vector of a state as input and returns the action that should be executed within that state. This will be the action that will yield the expected average reward maximization. Reinforcement learning is very general and can be applied to a particular kind of problem in which decision-making is sequential and the goal is long-term, such as game playing, robotics, resource management, or logistics (Linares et al. 2023).

Apart from supervised and unsupervised, there is also an additional approach in machine learning: semi-supervised and self-supervised learning. Semi-supervised learning deals with a hybrid model that involves training the algorithm from a little labelled data and a lot of unlabelled data. It adopts the best of both supervised and unsupervised learning methods to upskill learning precision with less set of manually labelled data. Self-supervised learning, on the other hand, is the method where the system learns to predict part of the input from other parts of the input while only utilizing unlabelled data. This approach allows, in a sense, for the model to understand the structure of the data by teaching itself from the data, so it can be very useful for those kinds of tasks where there is not so much labelled data, and this data is rare or expensive to get. Both methods aim at reducing the reliance on large volumes of labelled data, something that has been one of the major predicaments for machineries of learning (Linares et al. 2023).

Figure 9 showcases all the previously mentioned categories.

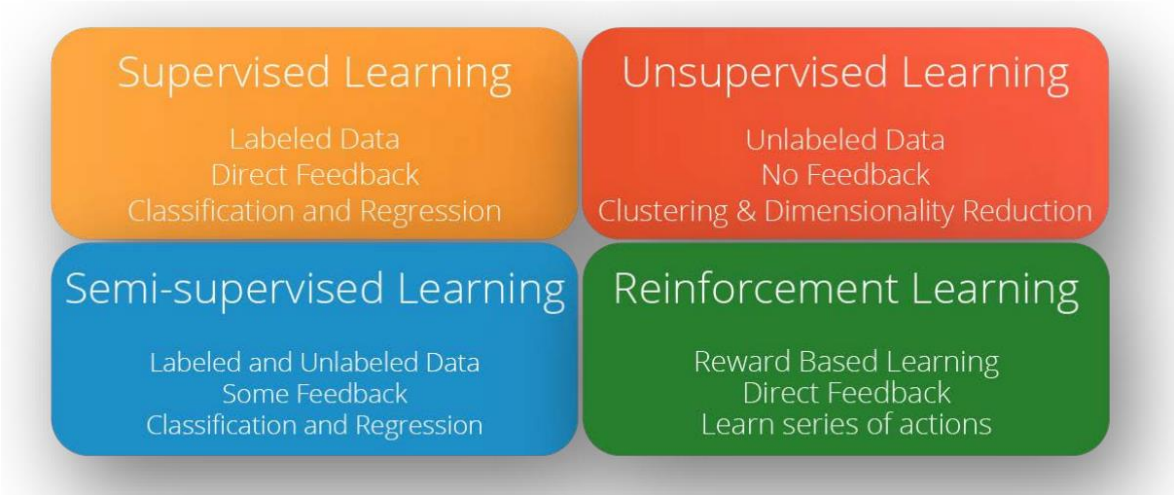

Figure 9. Introduction to Machine Learning. (Yalçın 2021)

Error Metrics: Overfitting and Underfitting

Evaluating the effectiveness of a model is a crucial step that must be prioritized from the outset of its development. This evaluation involves distinct metrics for supervised and unsupervised learning models, which are vital before the model's deployment. In supervised learning, a training set is utilized comprising examples with known categories or values to assess the model's performance. However, relying solely on these metrics with training data can lead to overfitting, where the model excels at predicting the training data but fails with new inputs. Therefore, adopting a best practice involves setting aside a portion of the data as a test set, not used in training, to gauge the model's real-world accuracy and ensure it generalizes well to unseen data (Linares et al. 2023).

As mentioned before, overfitting is the point at which a model has been trained excessively with its training data and hence captures, besides the underlying patterns, even the noise or irrelevant detail. Overfitting compromises the ability to generalize the model. It will be very bad in prediction or classification of new, unseen data. Machine learning models aim for generalization, the capacity to apply insights gained from a training dataset to novel scenarios. Overfitting generally comes from overly complex models or models that have been trained too long. In other words, a model will work very well on its training data but will perform very poorly on new data. Indicators of overfitting include low error rates on training data but high error rates on unseen test data. To combat overfitting, part of the data is reserved as a test set to evaluate the model's generalization ability, with disparities in performance between the training and test datasets signalling the occurrence of overfitting (IBM c.).

In contrary, underfitting occurs in data science when a model is too simplistic to accurately capture the relationship between input and output variables, leading to high error rates on both training and unseen data. This can happen due to insufficient training time, inadequate input features, or excessive regularization. Such models fail to identify the main trends in data, resulting in training errors and poor performance, and are characterized by high bias and low variance. Unlike overfitting, underfitting models are easier to detect as they struggle to generalize to new data, undermining their utility for classification or prediction tasks in everyday applications of machine learning algorithms (IBM d.).

The challenge now lies in balancing the model to avoid both overfitting and underfitting, aiming for a "sweet spot" that allows the model to accurately generalize from training data to new situations, as it is shown in Figure 10 (IBM c.).

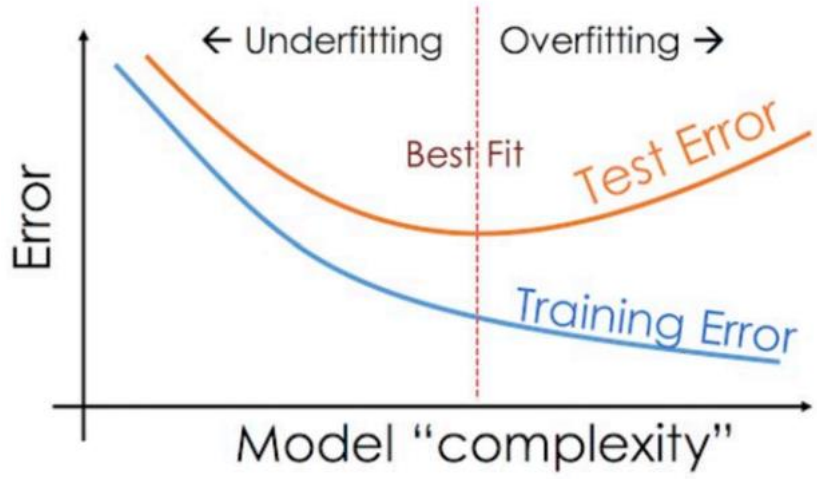

Figure 10. Underfitting vs Overfitting (Guetari et al. 2023)

Bias and quality of data

Bias in machine learning models, often manifesting as inappropriate or non-objective predictions, stems not from the algorithms themselves but from the data used for training. This demonstrates the fundamental importance of high-quality data in ethical sourcing not containing bias related to geography, ethnicity, religion, or other sensitive factors able to screw up the predictions. Proper data collection is one of the main steps to take care of, because the volume and diversity of data play an important role in methods such as deep learning, meaning the models must be trained on very comprehensive and balanced datasets. This ensures that the data is balanced since models without balanced data will have a heavy tendency leaning toward the categories which have a large representation. Among the basics in the data preparation for training include things like missing values,

scaling features, and encoding of categorical data to numeric formats. It further demands continuous monitoring and update of new, diverse datasets for fairness and accuracy, hence pointing toward the dynamic nature of maintaining machine learning models (Linares et al. 2023).

3.5 Deep Learning

Shown in Figure 11, deep learning is a specialized area within machine learning that focuses on leveraging algorithms modelled after the brain's architecture, known as artificial neural networks. This field's development hinges on the availability of high-speed computing resources. Although Deep Learning models draw loose inspiration from the ways in which biological nervous systems process and relay information, they significantly differ in both structure and functionality from actual human brains (Yogasudha, 2021).

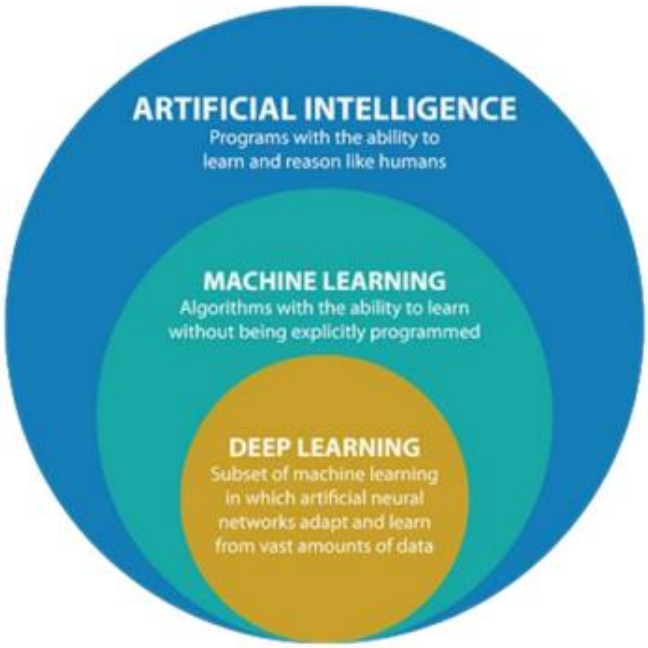

Figure 11. Deep Learning (Yogasudha, 2021)

Starting from the 1980s, there arose an interest in neural networks, which was highly driven by the connectionism movement. It was based on the assumption that many small computing units or artificial neurons' networks can give rise to intelligent behaviour through their interconnection. This was the time backpropagation algorithm was being developed that would form the base for the needed training of neural networks. Though with this development of sequential Long Short-Term Memory (LSTM) networks, Support Vector Machines (SVMs) were favoured over many other applications through the whole of 1990s. Nevertheless, some serious breakthroughs had been made in the beginning of the 2000s by those pioneers like Geoffrey Hinton, Yoshua Bengio, and Yann LeCun, who

had furthered the boundaries of the possibilities of neural networks. That laid the foundation for what's become the revolution in deep learning; real breakthroughs have been made since 2012, with the advent of AlexNet. This period has brought out a critical shift towards the deep learning models, characterized by the depth and complexity of the network that is behind the outlandish improvements in performance across tasks and has firmly entrenched deep learning as the dominant force in AI research and applications (Linares et al. 2023).

The Artificial Neuron

The concept of the artificial neuron was first proposed in 1943 by McCulloch and Pitts, known as the Threshold Logic Unit or Linear Threshold Unit. Then, in 1958, Frank Rosenblatt introduced the idea of the perceptron, which is a type of binary classifier (Linares et al. 2023).

As Figure 12 showcases, it consists of multiple inputs (a1, a2, a3, ... an) each paired with corresponding weights (w1, w2, w3, ... wn), and a bias term (b). The bias acts as a form of predisposition, introducing a constant and input-independent value that influences the neuron's output. These components are integrated through an activation function, which operates on the sum of the weighted inputs and the bias. This structure allows the neuron to perform complex computations, enabling it to contribute to the network's ability to model non-linear relationships and patterns within data (Linares et al. 2023).

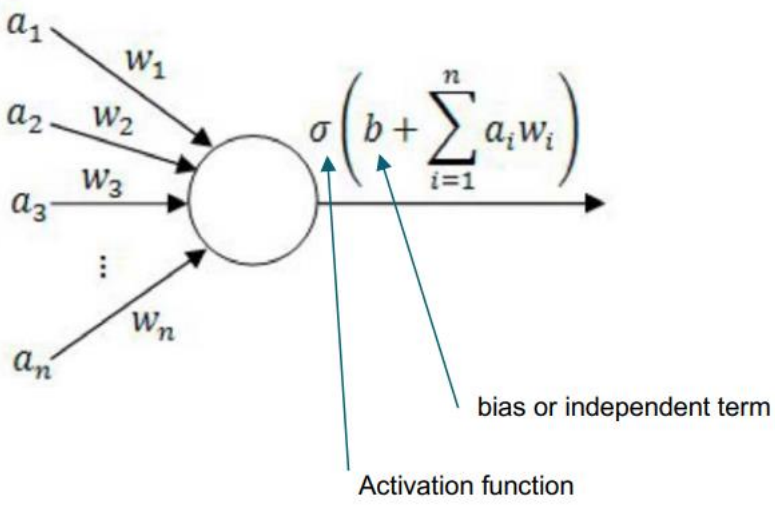

Figure 12. The Artificial Neuron (Linares et al. 2023)

Neural Networks

A neural network is a type of machine learning model intended to replicate the decision-making process of the human brain by interconnecting nodes or artificial neurons in general, otherwise known as an artificial neural network (ANN) or a simulated neural network (SNN), shown in Figure 13. The layers within this kind of network are an input layer, several hidden layers, and an output layer, where each node has its weight and threshold. Once the sum crosses a certain threshold, output from the nodes is activated, with the data passing to the next layers while making complex decisions based on input data. Neural networks have a learning phase with the help of training data aimed at increasing their accuracy. They are hence best applied to exercise like speech and image detections, among others, with the aid of data processing time that is much lower compared to human manual efforts. The Google Search Algorithm is one of the most outstanding uses of technology in neural networks, proving again the far-reaching influence of ANNs in computer science and AI fields through rapid classification of data and improvements to clustering at the core of deep learning (IBM e.).

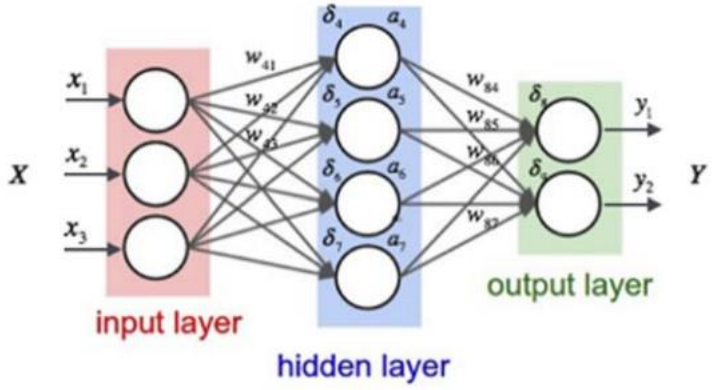

Figure 13. Neural Networks (Creative Commons Attribution 4.0 International)

Neural network training begins with the random initialization of weights, which represent the significance of parameters to the neuron's function. The accumulation of all weights and biases through learning constitutes the neural network's knowledge derived from the dataset. During training, input data is processed—weighted, and passed through activation functions to generate outputs. The backpropagation algorithm then adjusts the weights and biases in the opposite direction of the gradient, using gradient descent as a crucial tool for optimization. This iterative process, involving numerous examples, continues while regularly assessing accuracy until the desired performance is achieved (Linares et al. 2023).

3.6 Foundation Models, from LLMs to LMMs

Foundation Models represent a groundbreaking advancement in AI, leveraging deep learning and being trained on extensive, often unlabeled datasets. These models signify a leap forward, capable of mastering a broad spectrum of tasks in various fields without direct training for those specific tasks. They possess the remarkable ability to adapt or be fine-tuned using a minimal amount of labeled data for particular tasks, showcasing what is known as "in-context learning." This capacity allows these models to infer and carry out tasks based on simple prompts and a handful of examples, exhibiting new behaviours unseen in smaller or less diverse training sets. Foundation Models are pivotal in the ongoing evolution of AI, expanding the limits of machine learning's capabilities. However, they also prompt critical discussions about the implications of data standardization, the concentration of power, and the broader socio-technical challenges they introduce (Schneider 2022).

Natural Language Processing (NLP) and Language Models

Natural Language Processing (NLP) represents a critical facet of AI that focuses on enabling computers to understand, interpret, and respond to human language in a manner that is both natural and intuitive. This field integrates a complex blend of computational linguistics and computer science, utilizing both rule-based and machine learning algorithms to process and analyse large amounts of natural language data. The importance of NLP lies in its diverse applications, ranging from voice-activated assistants and chatbots to text analysis and language translation services, which enhance interaction between humans and machines, streamline business operations, and facilitate access to information. Over the years, NLP has evolved significantly, moving from simple rule-based approaches to sophisticated machine learning and deep learning techniques, enabling more accurate and contextually relevant interpretations of human language. This evolution underscores the continual growth and adaptation of NLP technologies to meet the ever-changing demands of linguistic data processing in various sectors, including healthcare, customer service, and content generation (Gillis).

Within the domain of natural language processing (NLP), language model generally refers to a basic framework that predicts the likelihood of occurrence for the following word given the context presented by the preceding words in the sequence already considered in the model. This effectively estimates the probability distribution of the word sequences, to naturally foresee which word would follow a given set of words. For example, given a sentence "The cat sat on the," then the model would most likely predict the next word as "mat," naturally against "apple" or "sky," which would get negative scores as alternative

word completions. This predictive capability is useful not only for text generation but, in addition, increases the power to understand texts by comprehending the structure and sequential logic of language (Linares et al. 2023).

The training of language models primarily employs an unsupervised method known as autoregressive language modeling. In this way, the model does learn to predict the next word in a sentence, even without having explicitly labeled data but rather is trained on a big corpus of text to induce patterns and relationships between words. During the training, it reads sequences of words and tries to predict the next word. In the process, the internal parameters are getting adjusted based on the accuracy of previous compared with actual words that followed afterwards. All these processes are fine-tuned iteratively for many sequences using a sliding window technique in the selection of sequences to further enhance and increase the ability of the model for better prediction of the target structures. Besides, the level of novelty in the text generated is further adjustable by using an increased or decreased 'temperature' for the model, a parameter that allows one to modulate the predictability of the output text and hence include controlled creativity or randomness while generating the text (Linares et al. 2023).

Word Embeddings

Word embedding represents a pivotal advancement in NLP, enabling the encoding of text into real-valued vectors that significantly enhance computer comprehension of textual content. By mapping words and documents to numeric vectors, this technique ensures that words sharing similarities in meaning are represented by proximate vectors, thus preserving essential semantic and syntactic characteristics. The transformation of text into such numeric representations allows NLP algorithms to efficiently process and understand textual information, marking a crucial breakthrough in leveraging deep learning to address complex NLP challenges (Turing).

Vectors are a much better sort of representation. In the context of words, it's convenient that each dimension of the vector could be a semantic feature. This could give us very interesting possibilities. Figure 14 shows how these features are selected to represent the words (Linares et al. 2023).

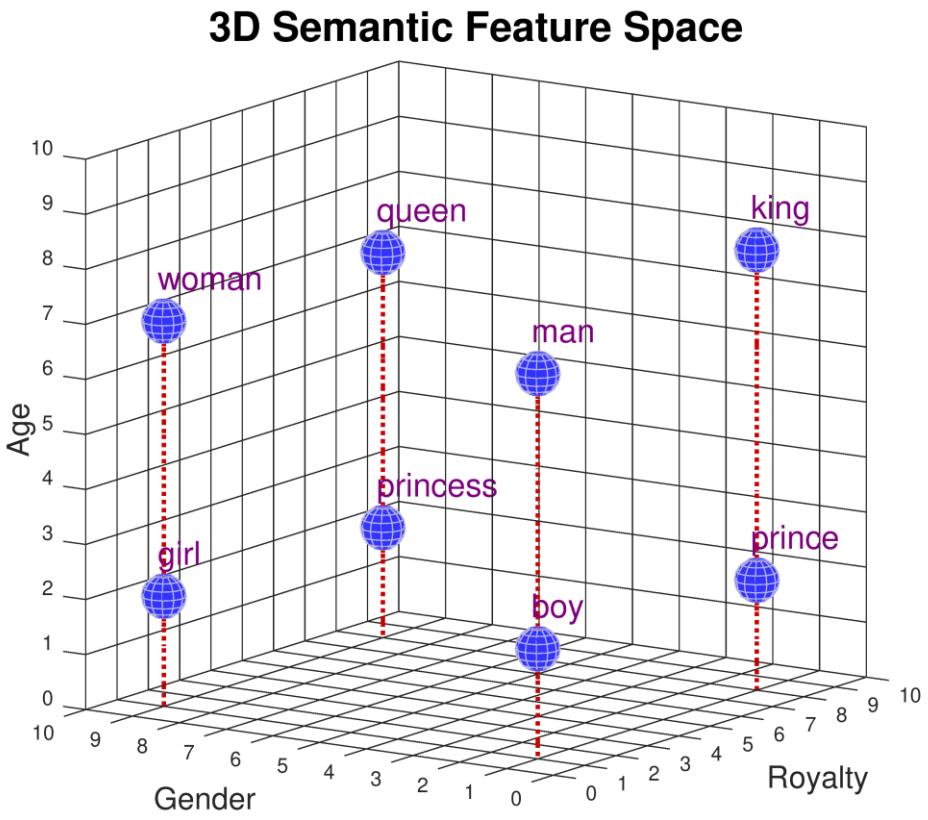

Figure 14. 3D Semantic Feature Space (Carnegie Mellon University)

However, if letting the model decide, which is among the desired properties of a nice feature, the human factor would be removed in this decision for the time being. In turn, it is well known that the number of dimensions, or why not semantic features should be of the order of a few hundreds. So, use a NN will be used to find these features, and do that by using a large corpus. The techniques used here are a bit complex, with word2vec being one of the best-known techniques. Somehow, these methods take into account the co-occurrences of words in large corpora. But when dealing with the whole corpus, similar semantic words are put in a similar region of the hyperspace or semantic features. So, finally, words with semantically similar properties will be mapped in similar regions or areas of the hyperspaces of word embeddings (Linares et al. 2023).

Large Language Models (LLMs)

Large language models (LLMs) represent advanced deep neural networks meticulously designed to excel in various natural language processing (NLP) tasks, including but not limited to, language translation, text generation, and summarization; as seen in Figure 15. The defining feature of these models is their extensive parameter count, empowering them to discern and replicate complex patterns and relationships within textual data. This extensive parameterization not only enables LLMs to manage extensive vocabularies with ease but also to produce text of remarkable quality (Linares et al. 2023).

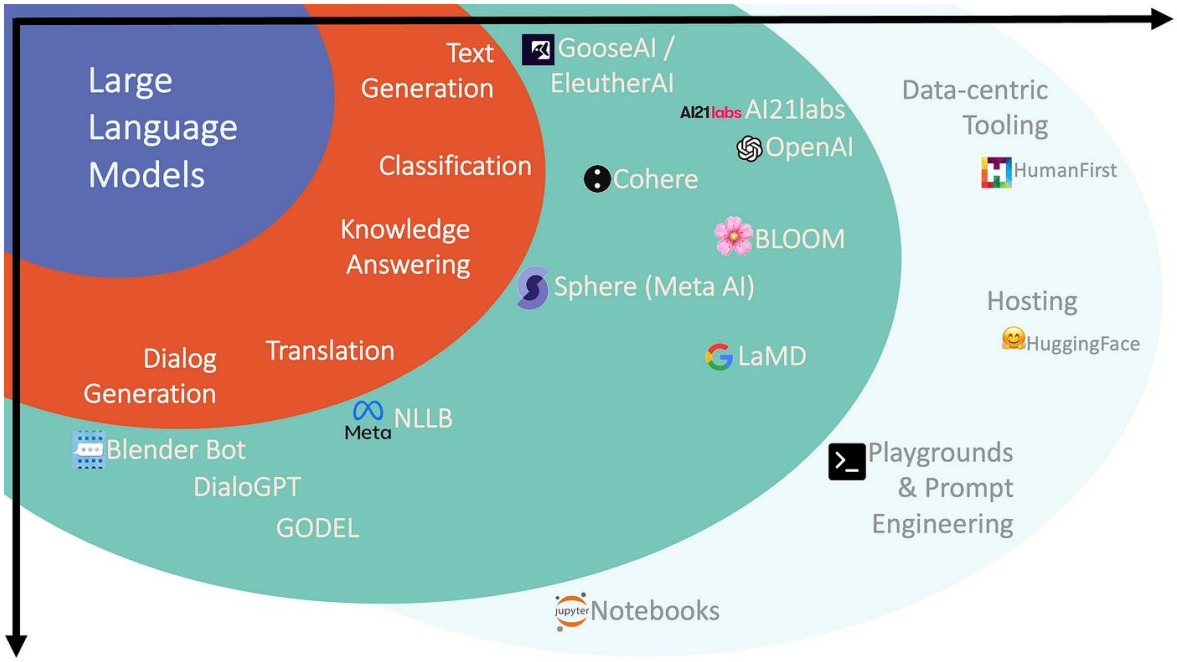

Figure 15. The Large Language Model Landscape (Greyling 2022)

Interestingly, LLMs exhibit what is known as emergence abilities. This phenomenon, where models display an aptitude for acquiring new skills and solving tasks they weren't explicitly programmed for, such as reasoning, arithmetic, and learning from a few examples, was initially unexpected. Renowned examples of LLMs include GPT, BERT, RoBERTa, T5, and BLOOM, with GPT (Generative Pre-trained Transformer) by OpenAI being a standout for its vast neural network architecture. Through additional fine-tuning, including supervised learning and Reinforcement Learning with Human Feedback (RLHF), these models can be further tailored to achieve exceptional accuracy in specialized tasks and domains (Linares et al. 2023).

Large Multimodal Models (LMMs)

Large Multimodal Models (LMMs) are at the forefront of AI, equipped to process and interpret multiple types of data simultaneously. Unlike conventional AI systems that are designed to work with a single data modality, such as text or visuals, LMMs are adept at handling a variety of inputs including text, images, audio, and video. This versatility allows them to understand and analyze information in a more comprehensive manner, mirroring the multifaceted way humans interact with the world (Santa María Morales 2024).

The essence of LMMs lies in their multimodal capabilities, enabling them to integrate and make sense of diverse data forms seamlessly. This ability not only enhances their performance in tasks like NLP, image and speech recognition but also empowers them to deliver human-like responses in applications such as chatbots and virtual assistants. The development and deployment of LMMs signify a pivotal shift towards achieving more sophisticated and human-like AI, offering a nuanced approach to how machines perceive and understand the complexities of the world (Santa María Morales 2024).

3.7 State of art

AI is amongst the rapidly evolving and growing areas, which have experienced quite many strides over the recent past. This has had great effects on various industries and is well placed to define their future.

GPT-4 represents a significant advancement in the field of AI, developed by OpenAI as the latest iteration in the Generative Pre-trained Transformer series. This multimodal model is designed to accept both text and image inputs, delivering text outputs with capabilities that approach human-level performance on a wide array of professional and academic benchmarks. Notably, GPT-4 demonstrates a remarkable improvement over its predecessor, GPT-3.5, especially in terms of reliability, creativity, and nuanced understanding of complex instructions. For instance, it achieves scores within the top 10% on simulated bar exams, a substantial leap from GPT-3.5's bottom 10% performance. This advancement is the result of a six-month iterative alignment process, drawing on lessons from adversarial testing and ChatGPT, which significantly enhanced its accuracy, steerability, and adherence to safety guidelines. Figure 16 showcases the exam results by performance of all the GPTs (OpenAI 2023).

The development of GPT-4 involved a comprehensive overhaul of OpenAI's deep learning infrastructure, in collaboration with Azure, to tailor a supercomputer specifically for this purpose. The model's training process was remarkably stable, allowing for accurate performance predictions, which OpenAI deems critical for future safety and reliability en-

hancements. GPT-4's design facilitates a wide range of applications, from improving customer support to assisting in programming tasks, by understanding and generating human-like responses. Additionally, its ability to process visual inputs alongside text opens new avenues for AI applications across various domains. Despite its impressive capabilities, GPT-4 maintains certain limitations common to AI models, such as generating factually incorrect information or "hallucinating" details, underscoring the importance of cautious application in high-stakes scenarios (OpenAI 2023).

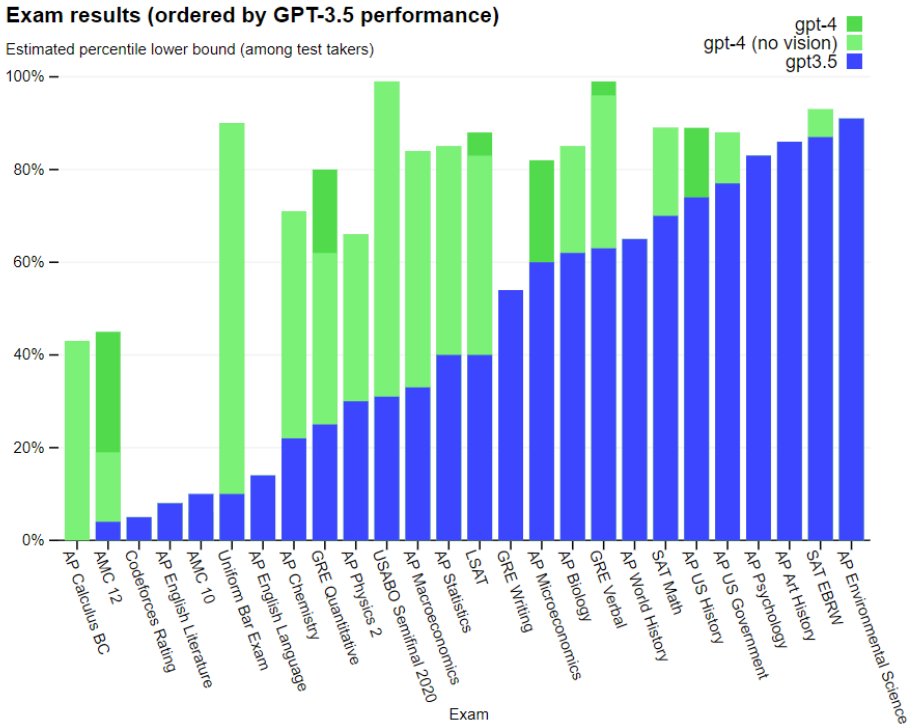

Figure 16. Exam results (OpenAI 2023)

DALL·E-3, the latest iteration of OpenAI's text-to-image generation model, introduces enhanced capabilities and features, enabling it to produce high-quality images across diverse domains with remarkable attention to detail and creativity. This version maintains the familiar API endpoints from its predecessor, DALL·E-2, while introducing significant improvements such as prompt rewriting with GPT-4 for optimized input processing, the introduction of 'natural' and 'vivid' styles to cater to different aesthetic preferences, and a new 'quality' parameter allowing for 'standard' and 'hd' image generation. DALL·E-3 also supports a wider range of image sizes, improving flexibility in output and aspect ratio for applications ranging from professional photography to digital art. These advancements not only expand the creative and practical applications of the model but also streamline the process for generating custom icons, logos, tattoos, stickers, and more, opening new possibilities for personalized content creation and design (Depue 2023).

4 Development tools

4.1 Front End

Front-end development is the aspect of web development that focuses on creating the user interface and experience of a website. It involves the use of programming languages, design principles, and tools to ensure that website visitors can seamlessly interact with and navigate through a site. Front-end developers are responsible for the visual presentation of a website, including its layout, colours, fonts, and overall design (Simmons 2023).

The significance of frontend development in the realm of computer science is highlighted by its essential role in converting concepts into actual products and services (Barroca 2017). It plays a pivotal role in managing input languages and benefits significantly from the incorporation of information technology to bolster innovation efforts (Gordon 2008). Moreover, developing knowledge-based frontend systems can prolong the operational life and improve the usability of critical software (Edmonds 1992). Collectively, these studies underscore the critical role of frontend development in promoting innovation, enhancing user interactions, and boosting the overall functionality of software systems.

Languages

• HTML: HTML is an acronym for HyperText Markup Language and is a kind of building block used in the making of web pages. It's a standard markup language used by web browsers to present content, from layout to even fonts, colors, and multimedia integration. Noted for its prolific use that could form the basis of close to 93% of all internet sites, HTML is esteemed for its simplicity and is even accessible to novices. The biggest limitation of HTML is that it is static, therefore an indispensable language in web development. HTML does require accompanying partners, CSS for styling, and JavaScript for the interactivity of a modern, dynamic web experience. But still, HTML remains the basic building block of web development and cannot be disputed in this matter. It stands to reason that a front-end developer must master it (Noble Desktop 2024).

• CSS: CSS (Cascading Style Sheets) is a powerful language that enables a web page to have style and layout. By separating design elements from the HTML structure, it allows developers to produce a website that is attractive, friendly, and visitor-inducing much easier. Its importance is because styles are applied across several pages; hence consistency is guaranteed, and much development time is saved. CSS, on the other hand, like everything else, has its own challenges to face. Foremost among them is its varying behaviour in different web browsers that can cause

arduous time in designing. However, it is critical for front-end developers to master CSS, as they are required to refine and turn the simplistic, banal, and unimpressive static HTML pages of a site into stylish, attractive, and dynamic web experiences that the end user will interact with (Noble Desktop 2024).

• JavaScript: JavaScript stands out as the programming language that infuses web pages with interactivity and dynamic content, elevating user engagement to new heights. Its ability to execute on the client side results in rapid interactions and immediate feedback, crucial for modern web applications. However, JavaScript's implementation can vary across different web browsers, presenting challenges in achieving consistent behavior. Additionally, its use in large-scale applications can lead to code bloat, impacting performance. Despite these challenges, JavaScript's ubiquity and its critical role in creating interactive web experiences underscore its indispensable value to front-end web development, making proficiency in JavaScript a key competency for developers (Noble Desktop 2024).

The combination of HTML, CSS, and JavaScript forms the foundational trio of technologies for web design and development, each playing a distinct and crucial role in building modern websites and web applications. Not only that, but they are the most popular amongst developers, as seen in Figure 17.

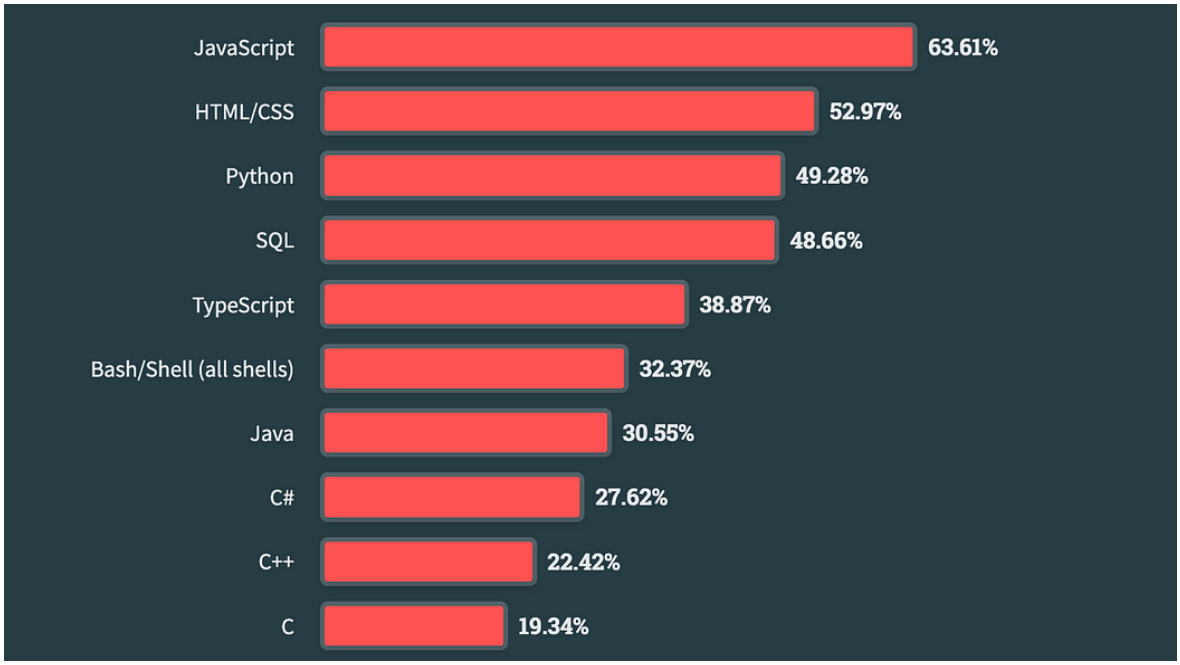

Figure 17. Most Used Programming Languages by Developers. (Shahid 2023)

Frameworks

Front-end frameworks have emerged as indispensable resources in the realm of web application development. They equip developers with a suite of pre-designed components, libraries, and utilities, facilitating the creation of dynamic and responsive user interfaces. The adoption of these frameworks not only enhances the performance of applications but also streamlines the development workflow, significantly reducing the time required to bring projects to completion. Through the provision of standardized, ready-to-use elements, front-end frameworks simplify complex aspects of UI development, enabling developers to focus on crafting unique user experiences (Mendes-Rodrigues 2024).

Here are some of the most relevant and cutting-edge frameworks being used nowadays:

• React: React, developed by Facebook, is more of a JavaScript library in that it allows a developer to change the way UIs are built with its component-based, declarative structure. It is JSX, which blends HTML with JavaScript, and a virtual DOM for fast updates that get React to be perfect for the construction of dynamic, large, and unordinary web applications, as Figure 18 showcases. React's ecosystem is rich with a vast array of third-party libraries and tools, bolstered by strong community support. This level of flexibility, combined with performance and a complete toolkit, makes this a go-to solution for projects across the board, from small startups to the most massive enterprise applications, an embodiment of the solution that strikes an equal balance between innovation and practicality (Mendes-Rodrigues 2024).

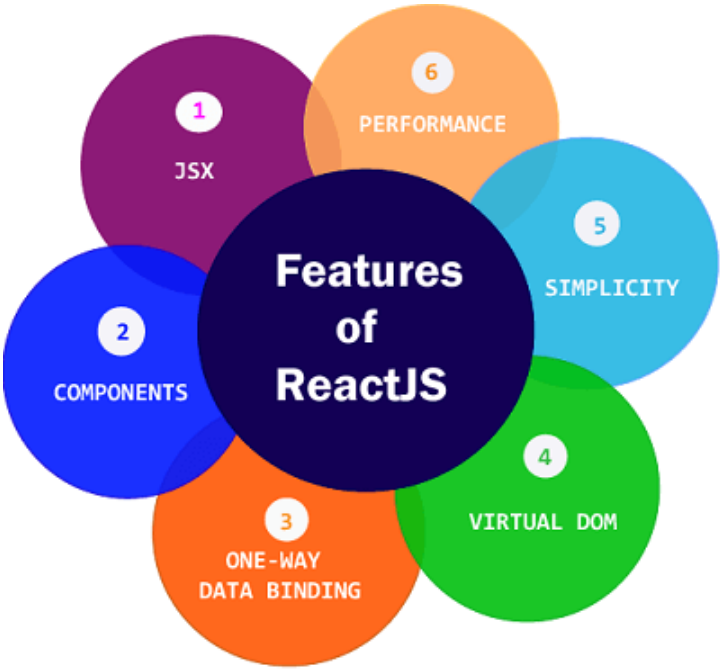

Figure 18. React’s Features (JavaTPoint b.)

• Angular: Angular, a product of Google's engineering brilliance, is a front-end framework that has consistently ranked high among developers for its ability to deliver rich, interactive web applications. It is built on TypeScript and comes equipped with a suite of tools and features designed to tackle the complexities of modern web development head-on. Angular's architecture encourages a modular approach, enabling developers to create highly scalable and maintainable applications. With features like two-way data binding, comprehensive routing, and an advanced command-line interface, Angular simplifies the development process while providing developers with the capability to build sophisticated, high-performance web applications. Its strong typing with TypeScript enhances code quality and readability, further bolstering Angular's position as a top choice for enterprise-level projects and developers who prioritize a structured development environment with robust tooling and support (Mendes-Rodrigues 2024).

• Flutter: Flutter is an open-source framework, designed and created by Google, which aims to help developers create visual parts of applications from a single code base for several platforms. Having only started in 2018 with a focus on mobile applications, in just a few short years, it has quickly risen to a point where it has already started supporting development across six platforms: iOS, Android, web, Windows, MacOS, and Linux. This evolution just shows the strength of Flutter in the building of interfaces for applications that must run fluidly over diversities of platforms, and that's exactly why this framework becomes highly popular among frontend and full-stack developers, who desire efficiency and consistency in the development of their UIs. Figure 19 shows a Flutter mobile application (AWS).

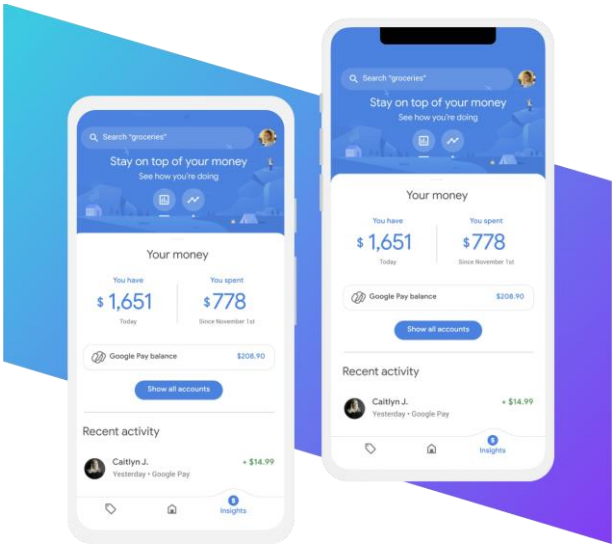

Figure 19. Flutter on Mobile (Flutter)

All these bring unique combinations, making Flutter stand out in the crowd. It makes app development easy across platforms. Unlike native app development, which is of high performance and complexity in developing, demanding greater resources and, inherently, platform-specific development, Flutter allows one to adapt to different platforms by using a single codebase. This, therefore, saves time and development costs in ensuring that the user experience on both platforms is uniform. It's built on the Dart programming language and its rendering engine, with developer utilities like hot reload and widget inspector. It further diminishes the appeal from the provision of near-native performance, easy UI building, and an attractive development experience (AWS).

4.2 Back End

Back-end development focuses on the server side of a website, encompassing the database, server, and application necessary for the front end to function. This area of development is concerned with the architecture and logic that drive a website's core functionalities, managing data exchange between the server and the users. Back-end developers are tasked with building and maintaining the technology that processes the data and operations of a website, ensuring everything on the client-facing side can exist and operate (Simmons 2023).

Server

Servers in computing are essential pieces of hardware or software that provide functionality for other programs or devices, following the client-server model. These servers offer various services such as sharing data, hosting websites, managing emails, and processing intensive workloads like database transactions. They operate within a network, where a client sends requests to the server, which then responds back with the required information or action. Servers can be specialized for different purposes like database servers, file servers, mail servers, web servers, game servers, and more, each catering to specific functions within the client-server architecture. Additionally, servers can be physical machines or virtual representations, with key components like processors, RAM, storage, and bandwidth crucial for their efficient operation (Kanade 2023; Posey).

The types of servers vary based on their functions, with common examples including web servers, mail servers, file servers, application servers, proxy servers, and more. Web servers host websites, mail servers handle email communicatioen, file servers share files over a network, and application servers provide business logic for applications. Servers like proxy servers act as intermediaries between clients and servers, while virtual servers

create a virtualized environment for multiple users. Each type of server requires specific hardware configurations to meet its demands, with considerations for fast processors, sufficient RAM, storage capabilities, and high bandwidth to ensure optimal performance and reliability in serving client requests (Kanade 2023; Posey).

API

API is an acronym that stands for Application Programming Interface. APIs nowadays form a very vital part of the software developed in that they allow giving and sharing services to other applications or software through communication and exchange of data between the applications but without necessarily being exposed to their implementation. In this sense, they will act as bridges for the interaction and sharing of services and information. The REST APIs are popular kinds of APIs due to the uniform interface, statelessness, and client-server communication, among other architectural constraints, thus being used widely as a web service. Basically, these are APIs built to make server-side data within easy reach through the service of simplified formats like JSON and XML, and so are efficient in communication over the internet (Eland; ITPro Today 2022). Figure 20 illustrates how this communication is done.

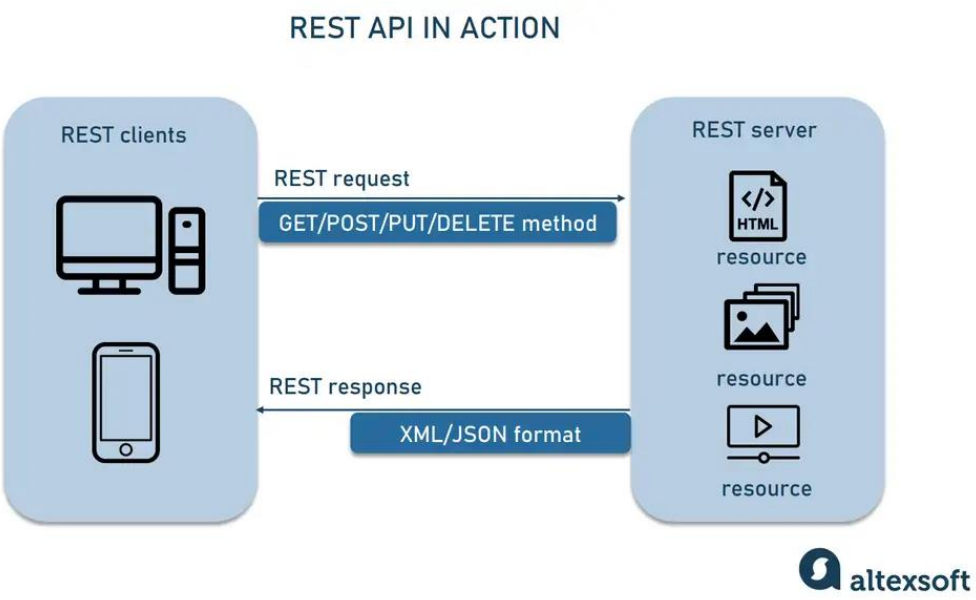

Figure 20. REST API in action. (Altexsoft 2022.)

Modern software development is anchored on APIs. Those are tools that connect services from diverse sources of information in a programmatically convenient way. APIs are applied in every industrial domain, from business to service delivery, to ensure a seamless digital experience regarding app-to-data connectivity. It is, therefore, an indication of just how APIs are being adopted, whereby a high percentage of organizations are integrating

both public and private APIs into their operations. Basically, RESTful APIs enable different programming languages and other models not to be based on the traditional request-response pattern, hence they are flexible, finding application in many other different contexts (Axway 2024; Acodez 2023).

The design principles of REST APIs are based on stateless behavior, where each request from the client to the server should contain everything that needs to be understood and, hence, consequently be served. This stateless, coupled with the use of standard HTTP methods of (GET, POST, PUT, DELETE), represents high scalability, and hence helps in the maintenance or increase of the independency that exists between client and server. That is, the server can change without having the capability of the client to talk back affected (Zhang & Arcuri, 2022).

More so, REST APIs form the very core of microservices architecture—a design approach to building software that structures applications as a collection of loosely coupled services. In this regard, REST APIs enable data interchange among such microservices, enhancing the modularity of applications and speeding up the development cycle with its maintenance and, perhaps, easier improvement in the quality of the results. Rest APIs are important for reliability and performance. Thus, effective techniques must be elicited for automated test generation (Kim et al. 2022).

Languages and Frameworks

Python is a versatile programming language favored for its readability and brevity, making it an excellent choice for both beginners and experienced developers. Frameworks like FastAPI and Flask leverage Python's strengths, offering robust solutions for web applications. FastAPI is recognized for its high performance and ease of use for building APIs with Python 3.7+ types, which significantly improves development speed and reduces bugs (Lathkar 2023; Bogachev & Zarikovskaya 2021). Flask, on the other hand, is a micro web framework that offers simplicity and flexibility, allowing developers to scale their applications with ease (Beregi et al. 2022).

Node.js has emerged as a powerful environment for executing JavaScript code server-side, enabling developers to build scalable and efficient web applications. It's particularly well-suited for handling asynchronous operations and real-time applications. Studies and developments in this area continue to expand, demonstrating Node.js's capacity for building robust web APIs and services (Zhang, Belhadi & Arcuri, 2023).

NestJS is a progressive Node.js framework designed for building efficient, reliable, and scalable server-side applications. It is built with and fully supports TypeScript, although it

also allows developers to code in pure JavaScript. NestJS leverages the powerful features of TypeScript, such as strong typing and object-oriented programming, to provide a level of abstraction above Node.js, thereby enhancing the development experience with higher security, scalability, and maintainability of applications. Drawing inspiration from Angular in terms of architecture and design patterns, NestJS introduces a modular organization of code, dependency injection, and an integrated system for handling HTTP server frameworks, which can significantly streamline the development of complex applications. Its architecture is aimed at creating highly testable, scalable, loosely coupled, and easily maintainable applications, making it an excellent choice for enterprise-level projects and microservices (NestJS.).

PHP, a server-side scripting language, remains a popular choice for web development due to its extensive database support and comprehensive array of built-in functions. It is widely used for creating dynamic web page content and server-side scripting. The integration of PHP with MongoDB exemplifies the modern approach to developing web 2.0 applications, blending the traditional PHP framework with NoSQL database technology to enhance application performance and flexibility (Islam 2011).

5 Practical Case: AI-Based Language Learning Application

5.1 Introduction

With the foundational concepts of language learning, AI, back end, and front end, it is time to transition from theory to application. This chapter delineates the development of an AI-based language learning application, illustrating the practical application of AI in enhancing educational technology.

The development of this application was motivated by the observation that most of the existing language learning applications, for instance Duolingo, do not sufficiently incorporate artificial intelligence, often resulting in a repetitive and monotonous learning experience. The result of such repetitiveness may be dullness in the learning process by the user who often goes through the same. Therefore, the application is built in such a way that it allows users with their smartphone's camera to ensure the learning is more fun and interactive. This approach not only improves the learning experience but also personalizes it, as users are allowed to explore language through real contexts of the world captured by their devices.

The utilization of AI in this language learning app is one of the rightful justifications to use, as it can bring up the level of interactivity and personalization of the learning activity, both being of core importance to keep learners efficient. The AI-powered speech recognition and image captioning also help the learners to get both pronunciation and vocabulary practice in real-world contexts through immersive activity-based practice. Thus, AI dives in making the process of learning deeply enriched, not making educational experience engaging but also tailor-made for individual learning style and needs diversification.

5.2 AI Selection

Before discussing further development of backend or frontend components of the app, there needs to be an agreement upon what AI models are going to be put into this application. This very much determines the architecture and design in frontend and backend, but also general functionality and the user experience of the app. Selecting the appropriate AI models involves an assessment of the specific capabilities needed, such as language recognition, image processing, and speech-to-text conversion. This will ensure that the adopted AI technologies are effectively being supporters of the interactive features within the app, giving real-time and exact analysis with feedback based on user interactions.

The choice of AI models also dictates the technical requirements and frameworks that will be implemented during development. It affects the scalability of the app, its responsiveness, and the efficiency of data processing. Therefore, making informed decisions about the AI models lays the groundwork for a robust and capable system. It sets a clear direction for the development process, helping to align technical specifications with educational objectives, thereby optimizing the learning experience tailored to user needs.

Open-Source vs Proprietary Models

An open-source model refers to software whose source code is made freely available and can be modified and distributed by anyone. This model is built on the principle of collaborative participation, transparent process, rapid prototyping, and community-oriented development (Community-Oriented Development 2014).

Proprietary models, on the other hand, are owned by companies, individuals, or entities who retain exclusive legal rights over the usage and distribution of the software. Unlike open-source software, the source code of proprietary software is not publicly accessible, ensuring that the control over software features, development, and distribution remains with the original creators (Kong et al. 2022).

For the application’s case, considering that several models are needed to accomplish the main functionality, OpenAI’s proprietary models are the ones being selected. The choice to use OpenAI's most powerful models for the design of the language learning app is majorly driven by the need for robust state-of-art AI functionalities, in which open-source models are simply no match. Such models, refined and further improved constantly, ensure nothing but the latest development in AI technology. The proprietary nature of OpenAI's models also guarantees a level of reliability and consistency essential for commercial application deployment, which can be challenging to achieve with open-source models that might lack uniform standards of quality and ongoing support.

Furthermore, using OpenAI over open-source models ensures access to comprehensive documentation and professional support, which is critical when integrating complex AI functionalities into the app. This support significantly reduces the time and resources needed for troubleshooting and fine-tuning, which can accelerate the development cycle and enhance the overall stability of the application. OpenAI's commitment to providing well-maintained and rigorously tested AI solutions means that the app can offer a seamless, efficient, and effective language learning experience, leveraging AI capabilities that are at the forefront of the technology.

Not only that, but OpenAI has developed a powerful API that incorporates its best models, including Whisper for speech recognition, DALL-E 3 for image processing, and GPT-4 for advanced text generation and understanding, as seen in Figure 21. This API offers a versatile suite of AI tools that are directly applicable to enhancing the interactive elements of the language learning app. By integrating these models, the app can provide real-time, accurate language translation, generate descriptive captions from images, and offer conversational interactions in multiple languages, making the learning process more engaging and effective.

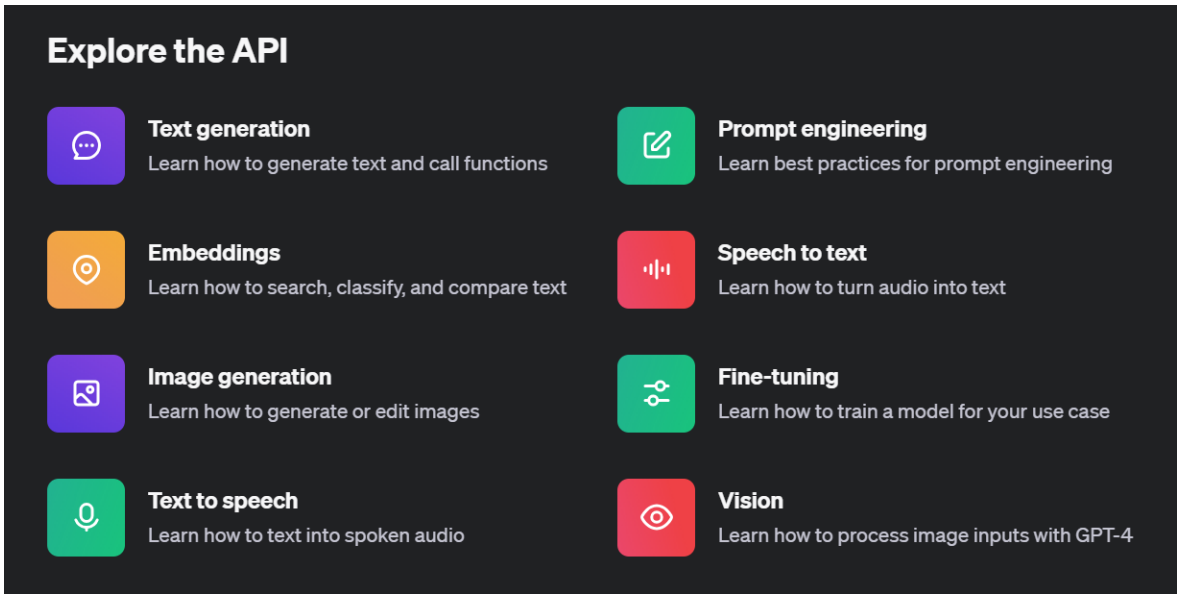

Figure 21. OpenAI's API models

Image to Text Model

The main functionality of the application will be to generate a text output from an image input that will be taken from the users' phone camera or gallery. Thus, it is clear that the application will require a model that is capable of generating text given an image. In this case, the model selected is GPT-4 Turbo from the OpenAI's API.

GPT-4 Turbo with Vision introduces the capability for the model to process and interpret images, enhancing its utility beyond text-based inputs. This feature enables the model to answer questions about the content and context of images by either providing a link to the image or passing the image directly in base64 encoded format. The model can handle multiple images simultaneously, adapting its responses based on the visual information provided (OpenAI API a.).

Speech to Text Model

Secondly, the application will have a pronunciation functionality, that is, users will be able to use their phone's microphone to record themselves pronouncing a certain word which previously would have been generated by GPT-4 Turbo. To achieve this feature, the application will need a speech to text model able to get a text output from an audio input. In this case, Whisper from OpenAI's API will be selected.

The OpenAI Whisper model is a powerful tool for converting spoken language into text, offering both transcription and translation capabilities. This model supports a wide range of languages and file types, making it highly versatile for various applications. Users can transcribe audio in the original language or translate and transcribe it into English, with support for common audio formats like mp3, mp4, and wav. The model can handle files up to 25 MB, suitable for many standard audio clips (OpenAI API b.).

For transcription, the model can output either raw text or a more structured JSON format with optional timestamps, providing flexibility in how the output is utilized, such as in video editing or detailed content analysis. The translation capability is particularly useful for multilingual environments, allowing users to convert non-English audio into English text efficiently. Overall, Whisper serves as a robust solution for a variety of speech-to-text needs, combining OpenAI's advanced AI technology with practical, user-friendly features (OpenAI API b.).

As explained, Whisper's capabilities in both translation and transcription successfully fulfil the needs of the application. Users will be capable of pronouncing words and phrases and will get immediate feedback from Whisper.

Text to Speech Model

Finally, users will want to listen to the correct pronunciation of the words generated by GPT-4 Turbo. This not only will be a useful feature but will also help users to pronounce accurately, improving the learning process. In this case, the model will not be selected from OpenAI's API, instead, ElevenLabs' API text to speech model is going to be selected.

ElevenLabs is a pioneering voice AI research and deployment company focused on breaking down linguistic barriers in multimedia content. Founded in 2022 by ex-Google and ex-Palantir engineers, Piotr and Mati, the company develops advanced AI-driven audio technologies capable of generating speech in multiple languages and voices. Its upcoming AI dubbing tool promises to transform content consumption by allowing users to re-voice audio and video in various languages while retaining the original speaker's voice

characteristics. Supported by leading investors like Andreessen Horowitz, Sequoia Capital, and prominent angel investors, ElevenLabs aims to enhance accessibility across education, entertainment, and real-time communication (ElevenLabs.).

The Text to Speech API by ElevenLabs allows users to convert text into lifelike speech across 29 languages using a vast selection of voices. This API, noted for its best-in-class latency and high-quality audio output at 128kbps, supports integration into various applications via Python libraries, React apps, or through websockets. Key features include ultra-low latency with ~400ms generation times, contextual awareness for nuanced speech, and a wide range of voice options accessible via the Voice Lab or the Voices Library. The API supports various output formats and offers detailed customization of voice settings per request, catering to diverse needs from simple voiceovers to complex AI-driven audio applications (ElevenLabs API.).

Choosing ElevenLabs to achieve the text to speech functionality was one of the best decisions, as it perfectly suited the needs of the application. Users now will be able to hear the correct pronunciation of any phrase or word in a vast array of languages, and in combination with the speech to text feature, will fulfil one of the main requirements of the application: to be multilingual.

5.3 Backend Development

Now that all the necessary models are clearly explained and justified, it’s time to integrate them into the backend of the application. The backend of the app is crucial as it handles core functionalities such as data management, processing user requests, and integrating with external services like the OpenAI API. It acts as the backbone that supports and connects the frontend with various resources necessary for smooth operation and user interaction. The backend ensures that data sent from the users' devices is processed efficiently, securely stored, and properly communicated back to the frontend for a seamless user experience. Thus, it is critical to evaluate each option available and select the most suitable one.

For this purpose, a RESTful API is a fundamental requirement. The backend for the application would require an API that is RESTful because of its mix of scalability, simplicity, and efficiency, by use with standard HTTP methods. This will ensure that more users are accommodated at the same time by the application without poor performance being experienced. The stateless nature of REST ensures a clear separation of concerns between client and server, facilitating independent development and easy integration with various platforms and external services. This architectural style enhances the interoperability of

the application in such a way that it can comfortably communicate with other REST-compliant systems and services, like AI processing tools. This is important for functionalities like speech-to-text and text-to-speech, making the application robust, flexible, and accessed across different devices.

For all the previous reasons, the natural process as developers would be to choose the programming language, framework, and possible libraries to build up the REST API. In this case, Python was the option for the programming language, while FastAPI was the framework which would integrate the RESTful API. Python is selected for its simplicity, readability, and the extensive support it offers through libraries and frameworks which are ideal for rapid development. It is also widely recognized for its robust performance in data processing, making it suitable for handling the complexities of AI-driven features such as image and speech recognition. In the other hand, FastAPI is chosen for its high performance and ease of use. It is one of the fastest frameworks for building APIs with Python 3.7+, thanks to its asynchronous support and automatic data validation capabilities. FastAPI also offers interactive API documentation automatically, which helps in testing and maintaining the API by developers and can significantly speed up the development process, as shown in Figure 22.

Figure 22. FastAPI automated documentation

After having decided all the back-end tools, a first test with FastAPI and Python was to be performed. The objective was simply to try running the back end using a local computer and build the application from there. Figure 23 showcases the basic look of a FastAPI application after installing all the necessary Python libraries. The server is initiated with the command "*uvicorn main:app –host 0.0.0.0 –port 8000*". This command effectively runs the FastAPI application defined in *main.py* on port 8000, accessible from any machine that can connect to the host machine over the network. The only purpose of this first step was to ensure the correct installation and initialization of the server, which would be dis-

playing a hello world message. The rest of the back-end development would follow afterwards.

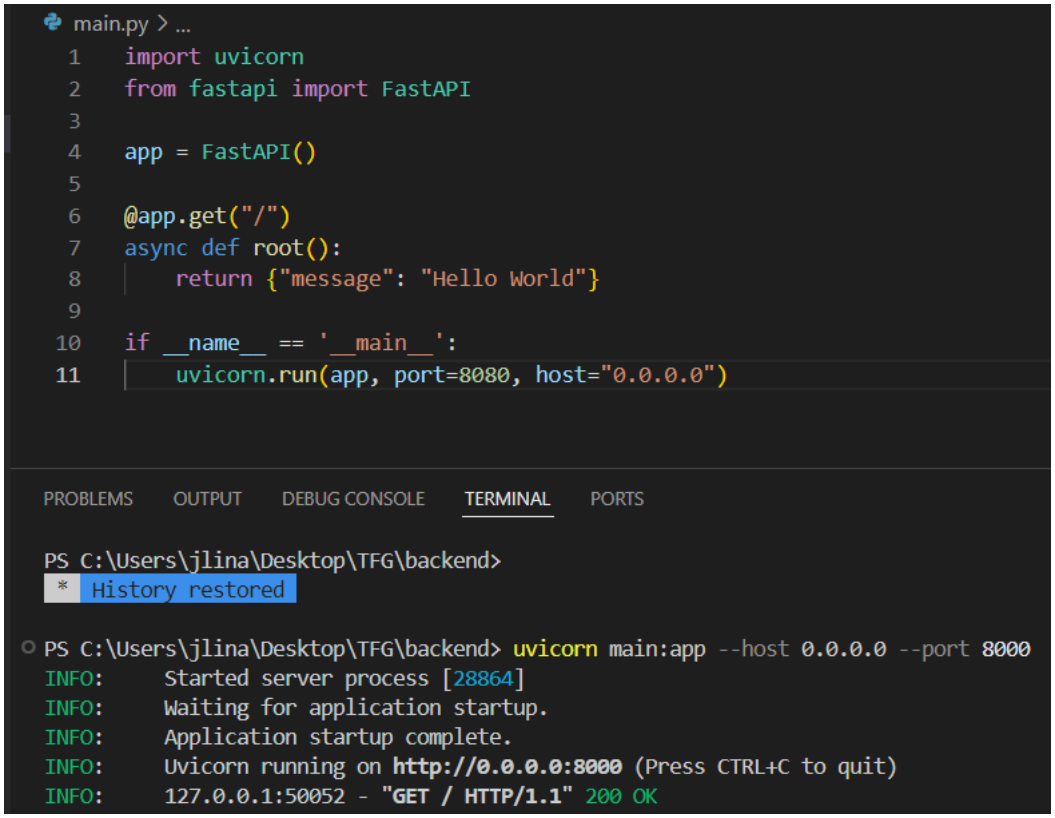

Figure 23. Backend with FastAPI correctly running

5.4 Frontend Development

With all the backend being set up at this point, it was time to decide the most important step of the whole practical case. This step is the mobile application development, for it is through the mobile application that the system interfaces with its users. Mobile application design, functionality, and performance are key since they have the direct greatest impact on the user experience and user satisfaction. The effectiveness of the application becomes an important factor that interfaces with users, an indispensable part of the entire project.

Such application serves to manifest these capabilities in an end-user facing front at this stage, where all the backend functionalities explained before having been developed. Importantly, it is a part of the collaboration to deliver a mobile application that would enable practical use of the backend features, increasing visibility and granting market penetration and user adoption access and ease to the target users of the developed product. Therefore, the developed mobile application influences direct effective engagement of the users, shaping the public image of the project. This takes even more importance when lan-

guage learning is being considered, as users will require an interface as much interactive and engaging as possible but keeping it at the same time simple and efficient as the objective is to learn a new language.

For the frontend development, Flutter was the chosen framework for the mobile application. One of the several reasons to choose it is that it can be able to bring forth high performance through a single codebase onto a couple of platforms. Flutter design is based on the use of widgets that contribute to simplifying the process of bringing coherent and good user interfaces together. This ensures the same user experience while using both the Android and iOS devices, hence saving the time and resources that would have been used in separative developments in the given platforms. Secondly, its performance has been said to be great because it compiles directly into native code, therefore needing no JavaScript bridge to communicate with native components of the device. The result is a much faster application startup time, combined with smoother animations that bring the whole user experience to new levels.

First Prototype

The next step to be followed is to create the first application prototype in order to test that all the development tools are working properly. The approach followed here was to connect both frontend and backend systems to make them work together. For that, a simple test application was developed. In this case, the user would be able to select an image either from the camera or gallery, and the application would recommend the user some words or phrases based on that picture. This implies that the model that was put under test was going to be the image-to-text GPT-4 Turbo model.

First of all, the connection between the backend and the OpenAI API was to be established. Figure 24 illustrates how this was achieved. This function, defined as an asynchronous endpoint the API, handles POST requests to generate textual descriptions of images uploaded by users. When an image file is received, the function first converts the image data into a Base64 encoded string. It then constructs a request payload containing this image data and sends it to the OpenAI API using the appropriate headers, including authorization via an API key. The request specifies the use of the "gpt-4-vision-preview" model and includes a default text prompt, which is the one that specifies what exactly will be extracted from the image, along with the encoded image data. If the request to OpenAI's API is successful, the function extracts and returns the textual description of the image from the API's response. This will be sent back to the frontend.

```python
# Call to Image-to-text, will return a text given the image
@app.post("/generate_text_description")
async def generate_text_description(image: UploadFile = File(...)):
    try:
        # Getting the base64 string
        base64_image = encode_image(await image.read())

        headers = {
            "Content-Type": "application/json",
            "Authorization": f"Bearer {api_key}"
        }

        payload = {
            "model": "gpt-4-vision-preview",
            "messages": [
                {
                    "role": "user",
                    "content": [
                        {
                            "type": "text",
                            "text": default_image_to_text_prompt
                        },
                        {
                            "type": "image_url",
                            "image_url": {
                                "url": f"{base64_image}"
                            }
                        }
                    ]
                }
            ],
            "max_tokens": 300
        }
        response = requests.post(openai_url, headers=headers, json=payload)
        image_description = response.json()["choices"][0]["message"]["content"]
        return {"generated_text": image_description}
    except Exception as e:
        raise HTTPException(status_code=500, detail=f"OpenAI API error: {str(e)}")
```

Figure 24. Image-to-text API call

Once this was done, the connection between backend and frontend was to be established. The frontend will need to call the FastAPI server in order to retrieve the text that the image-to-text function will return. This text is no other than the translated phrases the model had recommended from the image. Figure 25 shows exactly how this process is done in the frontend code, taking an image from the users' phone gallery. The function starts by setting a loading state to indicate that an operation is in progress. It then uses an image picker to let the user select an image. Upon selection of an image, the function constructs a POST request to the specified server URL (which is the IP of the FastAPI server in the local network), attaches the image file to the request, and sends it. Upon receiving a successful response (HTTP status code 200), the function decodes the JSON response to extract the generated text description of the image, updates the application state with this description, the image path, and any translations extracted from the description.

```dart
Future<void> sendImageToBackendGallery() async {
  try {
    setState(() {
      isLoading = true; // Show loading indicator
    });

    XFile? image = await _picker.pickImage(source: ImageSource.gallery);

    if (image != null) {
      var request = http.MultipartRequest(
        'POST',
        Uri.parse('http://192.168.43.56:8000/generate_text_description'),
      );

      request.headers['Content-Type'] = 'application/json';
      request.files.add(
        await http.MultipartFile.fromPath(
          'image',
          image.path,
        ),
      );

      var response = await request.send();
      var responseBody = await response.stream.bytesToString();

      if (response.statusCode == 200) {
        final dynamic responseBodyJson = jsonDecode(responseBody);
        final String generatedText = responseBodyJson['generated_text'];
        setState(() {
          description = generatedText;
          pickedImage = File(image.path);
          translations = extractTranslations(description);
        });
```

Figure 25. Sending the image to the FastAPI server and retrieving the output text

And after this process the only step remaining was to display all the translations alongside with the image in the frontend. The final prototype of the application allowed the user to choose an image either from the camera or gallery, getting a response in a bit more than five seconds. In this response the user could see a total of five translations alongside with each respective original meaning. In addition to this, each translation could be listened thanks to the text to speech functionality, which was implemented as well. All of it can be seen in Figure 26.

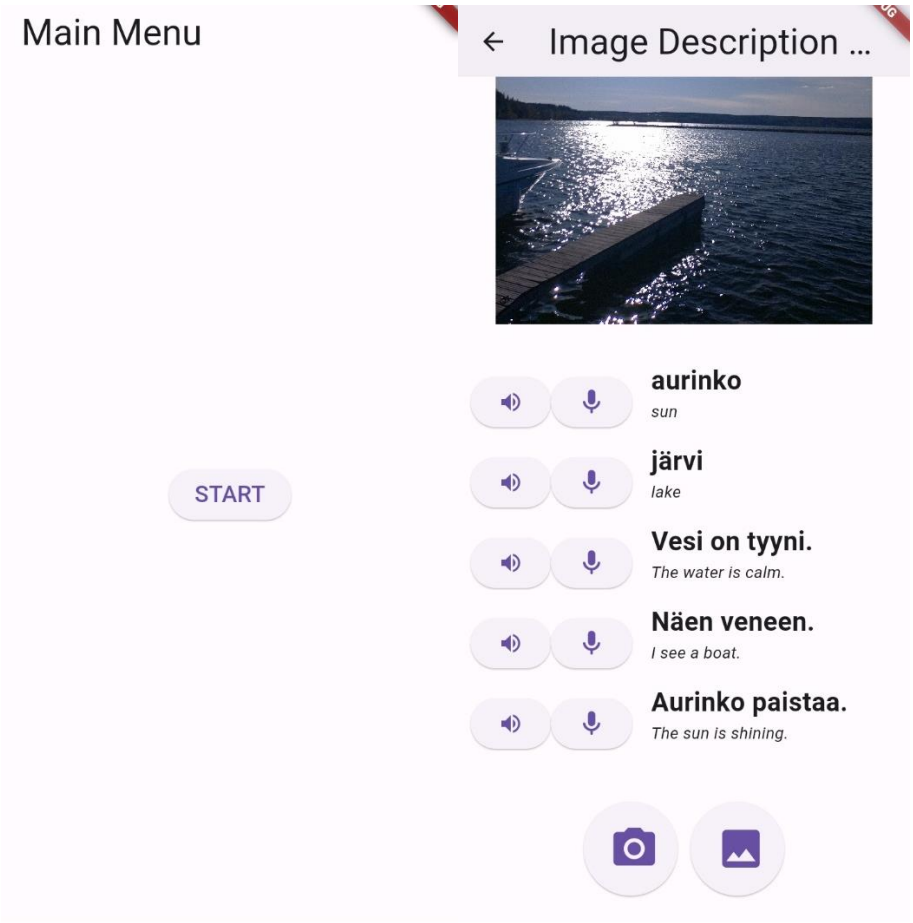

Figure 26. First look of the application

5.5 Final Application Development

After developing the initial prototype, the application already appeared quite promising, though there was still potential for enhancement. The current focus shifted toward refining various aspects of the application, adding a feature to save suggested words or phrases, integrating the pronunciation functionality, and making the application multilingual. Overall, the focus now was to improve the whole looks of the application, embracing simplicity and interactivity.

Local Data Persistence

The problem at this point was that there was no option for the user to save the translations for further review. To achieve this, in Flutter, several options are available for saving information depending on the application's requirements. For simple data like user preferences or settings, the SharedPreferences library is often utilized, which stores data in key-value pairs and is suitable for small amounts of data. For more structured or larger datasets, the sqflite plugin can be used, which provides access to SQLite databases. To store files directly on the device, the path_provider plugin allows access to the file system.

Additionally, for applications requiring cloud storage and synchronization, Firebase can be integrated, offering services such as Firestore for real-time data syncing and Firebase Storage for file storage.

The decision was to implement this feature using the *SharedPreferences* library in Flutter, as the information which must be saved is nothing else than an image with some texts, so there was no point to create a database for that. To achieve this, a model was created (seen in Figure 27), with the image path and three necessary string parameters (the original word, its translation, and the language).

```
class ImageData {
  final String imagePath;
  final String originalText;
  final String translationText;
  final String language;

  ImageData({required this.imagePath,
             required this.originalText,
             required this.translationText,
             required this.language});

  Map<String, dynamic> toJson() => {
    'imagePath': imagePath,
    'text1': originalText,
    'text2': translationText,
    'language': language,

  };

  static ImageData fromJson(Map<String, dynamic> json) => ImageData(
    imagePath: json['imagePath'],
    originalText: json['text1'],
    translationText: json['text2'],
    language: json['language'],

  );
}
```

Figure 27. ImageData model

This model is stored locally on the user's phone, allowing it to be accessed anytime. This enables users to save and later review the phrases or words they are interested in learning.

Speech-To-Text Integration

As this functionality was key for the application in terms of being a very useful tool for users to boost interactivity and simplicity, it was critical to ensure that it was developed properly. For this case, the *just_audio* library was used in Flutter. This library allowed the frontend to record audio from the user's phone. This audio was converted and later sent to the backend, which would be the one in charge of getting the transcription from Whisper, sending it back to the frontend. The final output text would be then compared to the original one, and checked, successfully achieving the pronunciation functionality.

Multiple Language Integration

One of the other benefits of the use of AI in the application is the capacity to operate in many different languages, which is a feature hard to achieve without AI (as it would require a lot of coding). This was also one key part of the application, as it would allow users from around the world to use the application in any country, and almost in every language. This functionality was fulfilled by sending one string parameter each time whenever the FastAPI server is called. This string contained the language flag in which OpenAI API had to return its outputs, so that all the results retrieved back would be in the language specified by this flag.

Final Application

Now that all the main functionalities were implemented successfully, the only thing remaining was to integrate them in the frontend, in the best way possible. This implied creating and designing an appealing user interface for the user, in order to boost user's engagement and interaction as much as possible, while keeping the application efficient and simple to use.

First of all it was clear the application needed a main menu, from which the user would be capable of navigating around the different features of the application. Figure 28 shows the final looks of this menu. It is composed by three main options: capture, review and pronunciation. These will be explained later, as they are different screens of the application on its own. With this separation, the user can easily select where to go, without further complications. Still, there was a feature missing on this menu, which was the multilingual feature. That's why on this screen a button is also provided to change the target language of the app, that is, the language the user wants to learn. One important point is that the application's base language will always be English. This was done to ensure that the only language that changes is the one that the user wants to learn, keeping the interface intact at any time. As seen in Figure 28 as well, the language button

leads the user into the language screen, where several example languages have been added to illustrate the case. It is important to remark that these languages shown are not the only languages that the application could support, as previously mentioned, the application could provide a large list of languages supported by AI's models. The languages shown are just some of the examples of the total amount. In this menu, the user can easily select the target language to learn, which will dictate afterwards all the models' language outputs.

Figure 28. Main Menu and Language Selector

Now diving into each of the options, the first option is the "Capture" functionality. This includes two ways of selecting an image by the user: camera or gallery (seen in Figure 29). After user selects an image from either of the mentioned options, the application processes that image and returns its output, which are five translations related to the image. Figure 29 also showcases this example, where all the translations and their original meaning are shown in the user interface, wrapped up in a scrollable UI component. This menu also allows the user to switch between three different types of image recommendations:

1. **Object Detection**

   The first option allows the user to display only object-related recommendations in the image. With this option selected, the image-to-text model will only show translations of the objects that appear in the image, typically being sole words.

2. **Question Formulation**

   The second option allows users to display only questions that they might have in the image context. With this option selected, the image-to-text model will only show translations of possible questions the user could ask in the image context.

3. **Image Description**

   The last option allows users to display the description of what is happening in the image. This is what the application displays by default when attaching an image from camera or gallery. With this option selected, the image-to-text model will only show translations of sentences describing the image.

For each of the five translations, the user will be able to hear the audio of any of them, thanks to the text-to-speech functionality. Finally, users will also be capable of saving any of the translations they consider useful, for further review. As explained before, this will successfully save the image and each translation in local storage.

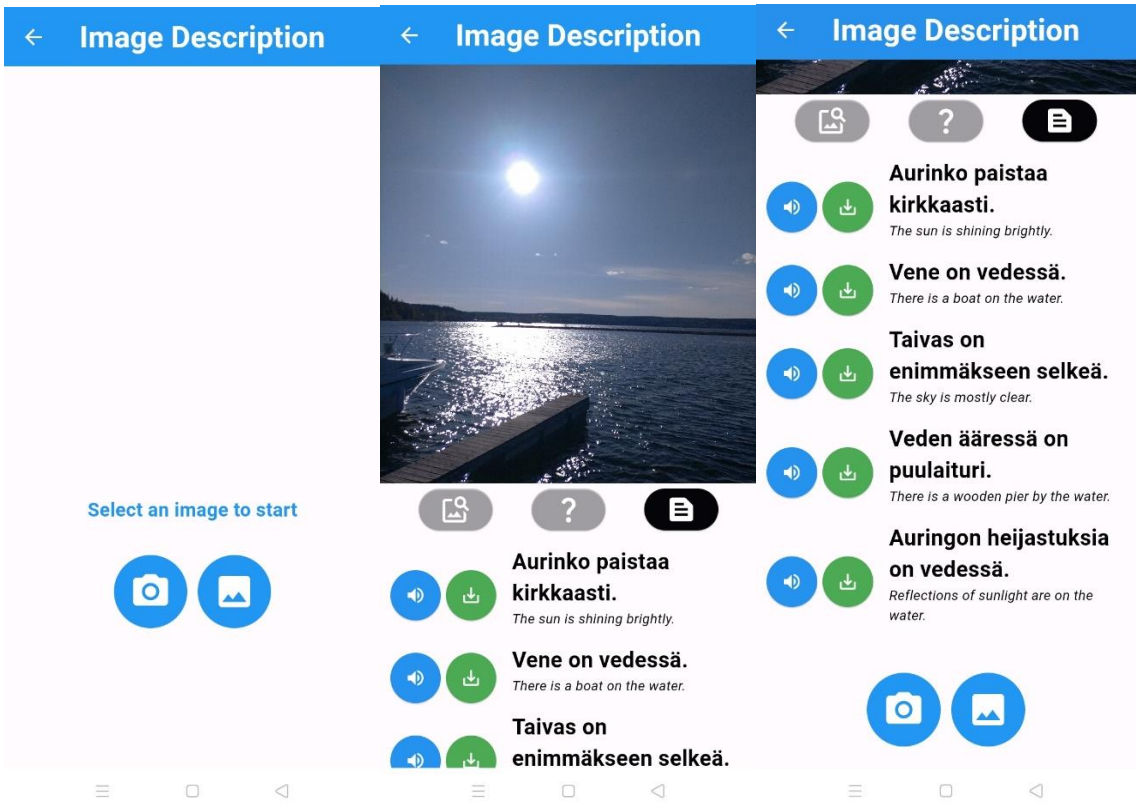

Figure 29. Image Description Screen

Secondly, there's the option to "Review" each of the translations saved previously in the capture menu. The approach followed in this case was based on showing each image and translation with a card view. This was inspired by another application used to study and memorize: Anki. Anki is a free and open-source flashcard program that uses spaced repetition to help users efficiently memorize and retain information (Anki.). With this approach, users can effectively study and memorize each of the cards containing both the image and the translation. Figure 30 illustrates one of the cards that were saved previously, and how the user can interact with it.

This screen contains a text field, where the user can input the translation of the phrase saved. For obvious reasons, the correct answer is not displayed until the user inputs an answer. When the user does so, an alert dialog will show on the screen, telling the user if the answer provided is wrong or correct. The user will also be able to hear the audio transcription with the text-to-speech functionality. To sum up, all the cards previously saved will be shown in this screen, providing an engaging and interactive way for the user to review each of the translations, while learning them at the same time. In addition, users will be able to delete any of the cards they want, by pressing the delete button at the bottom.

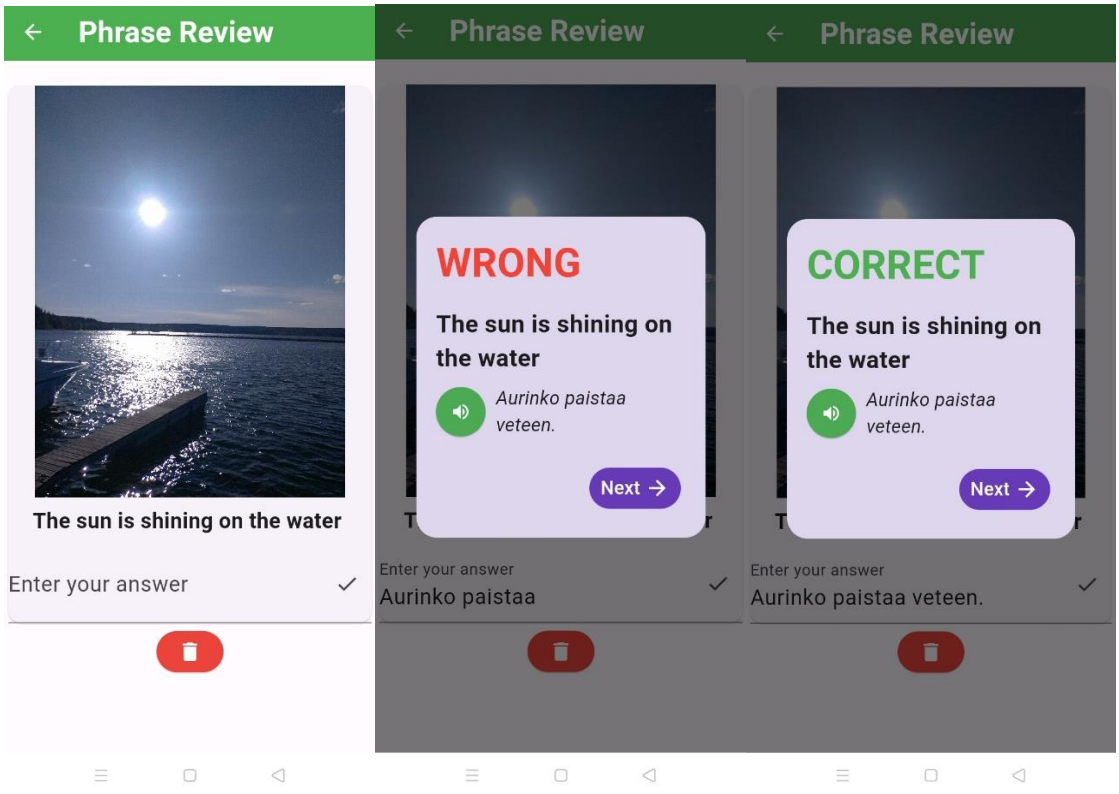

Figure 30. Review Page Screen

Finally, the third option is the “Pronunciation” functionality, where the speech-to-text feature is implemented. This screen follows the same card approach as the “Review” screen, where both image and translations are shown individually. The main difference is the fact that now users can record themselves pronouncing the translation. Figure 31 illustrates how this is done. Both the original and translated text are shown, with the text-to-speech functionality acting as a way to help the user replicate the pronunciation of each translation. Below, the user can make use of the speech-to-text by pressing the speak button and pronouncing the word or expression. Whisper will return the output text from the audio input of the user and this text will be compared with the translation, to see if they match or not. Then, an alert will be shown with the expected answer, and the answer output from the user, saying if this was wrong or correct.

This functionality emerges as one of the most innovative in the whole language learning environment, making the application both interactive and fun for the user to make use of it. Not only helps users to retain and learn languages, but also involves them in real time pronunciation, improving their oral ability in the target language.

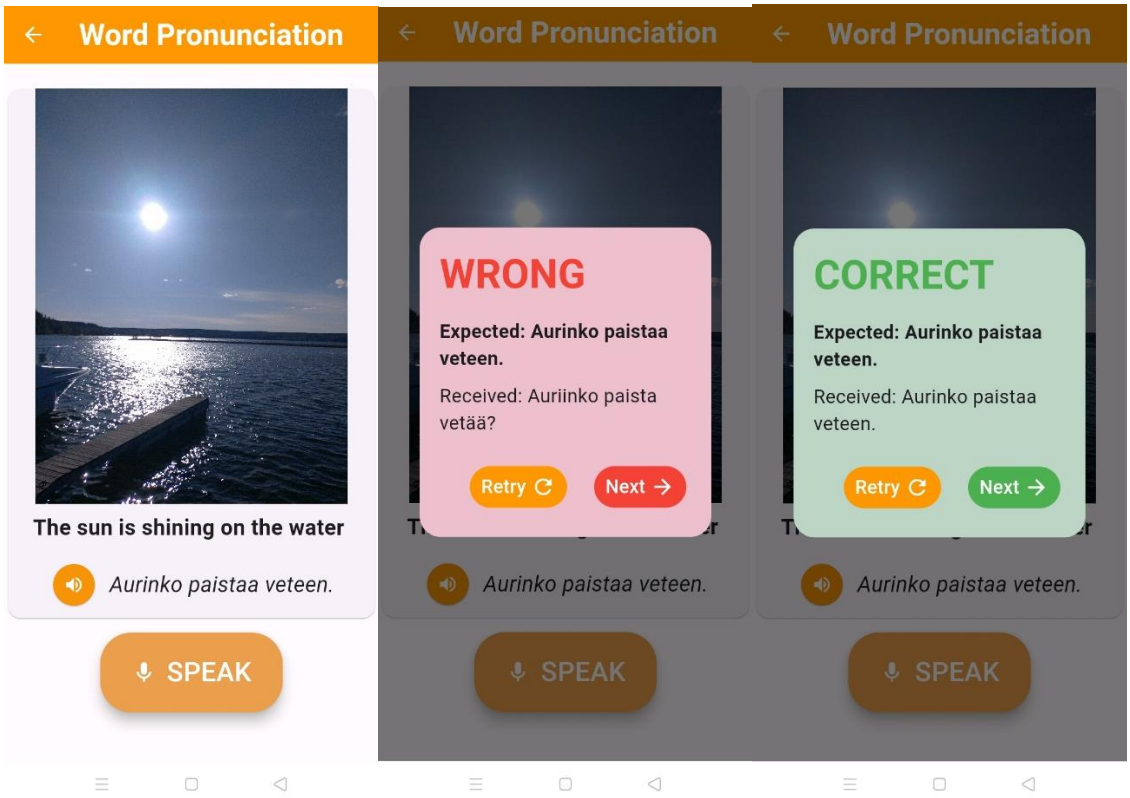

Figure 31. Pronunciation Page Screen

6 Conclusions

This thesis has provided a comprehensive exploration into the integration of artificial intelligence within the domain of language learning through the development of a mobile application. This synthesis of AI with language learning presents a significant step forward from traditional language learning tools, offering a more interactive, engaging, and efficient approach to language acquisition. By leveraging state-of-the-art AI technologies, including image captioning, speech recognition, and machine learning algorithms, the application delivers a unique and personalized learning experience that adapts to the individual's learning pace and style.

Throughout the thesis, we delved into various aspects of artificial intelligence, emphasizing its application in educational technologies, particularly for language learning. The theoretical discussions highlighted the transformative potential of AI in redefining traditional learning paradigms, emphasizing how AI can enhance the learning process by making it more engaging, interactive, and adaptable to the learner's needs. These discussions were not only foundational but instrumental in understanding the potential impacts and benefits of integrating AI into language learning.

The practical case of the AI-based language learning application demonstrated the application of these theories in a real-world scenario. The development process, from selecting appropriate AI models to designing the user interface, underscored the intricacies involved in creating an application that is both functional and user-friendly. The application's ability to use real-world images as a basis for language lessons represents a novel approach that situates learning within the context of everyday life, making it more relevant and practical.

Still, the final application obtained in this process isn't perfect at all. I faced some hard to solve problems when implementing and testing all the models. One of the several that still remain is the time the image-to-text model takes to suggest new translations. Since the API call has go through my backend, and then come all the way back to my frontend, it means that it takes more than seconds on average to get the output translations. It is not a critical issue but it's clearly one that could be improved in the future.

My intention is to further develop this application as I firmly believe it could be commercially viable. I still haven't seen another language learning application close to the one I have developed. One of the possible improvements would be to also include some way to detect objects or elements in the images, using a computer vision model. Another one would be to host the FastAPI server in some web service, to avoid having my own laptop acting

as a server. But these are problems for the future. Anyways, I believe I have achieved one of the objectives of the thesis, as I created a language learning solution that's both interactive and fun, solving one of the main problems of actual language learning applications like Duolingo, the excessive repetition which leads to boredom due to lack of AI integration.

I thoroughly enjoyed developing this thesis, most of it because it helped me learn Finnish along the way while creating a whole new application. I also learned how to put the premade AI models into practice and integrate them into a whole backend and frontend system.

References

Acodez. (2023). The Role of APIs in Modern Web Development. Retrieved on 8th April 2024. Available at: https://acodez.in/apis-in-modern-web-development/

Altexsoft. (2022). REST API: Key Concepts, Best Practices, and Benefits. Retrieved on 8th of April 2024. Available at: https://www.altexsoft.com/blog/rest-api-design/

Anki. About Anki. Retrieved on May 7th, 2024. Available at: https://apps.ankiweb.net/

Arnold, J. (2019). The Importance of Affect in Language Learning. Neofilolog. Retrieved January 29th, 2024. Available at https://www.semanticscholar.org/paper/THE-IMPORTANCE-OF-AFFECT-IN-LANGUAGE-LEARNING-Arnold/4b9dfb93a3c1d45aa693e3e93313eacf673d271d

Arunagiri, G., & Sumana, S. (2023). Modern Artificial Intelligence-An Overview. *September 2023*. Retrieved on 1st of March 2024. Available at: https://irojournals.com/aicn/article/view/5/3/4

AWS. What is Flutter? Retrieved on 25th of March 2024. Available at: https://aws.amazon.com/what-is/flutter/

Axway. (2024). Types of APIs: Different APIs Explained With Concrete Examples. Retrieved on 8th April 2024. Available at: https://www.techelevator.com/the-role-of-web-apis-in-modern-software-development/

Babbel (2023). Learn with your own voice: Babbel launches two new speech-based features. Retrieved on 26th February 2024. Available at: https://www.babbel.com/press/en-us/releases/learn-with-your-own-voice-babbel-launches-two-new-speech-based-features-us

Bali, S., & Liu, M. (2018). Students' perceptions toward online learning and face-to-face learning courses. *Journal of Physics: Conference Series, 1108*. Retrieved on 28th February 2024. Available at: https://iopscience.iop.org/article/10.1088/1742-6596/1108/1/012094

Barroca, J.P., Koria, M., Lindy, I., & Mulas, V. (2017). FrontEnd toolkit : a toolkit to transform ideas into intelligent action. Retrieved on 25th of March 2024. Available at: https://chooser.crossref.org/?doi=10.1596%2F26111

Bicknell, K., Brust, C., & Settles, B. (2023). How Duolingo's AI Learns what you Need to Learn: The language-learning app tries to emulate a great human tutor. *IEEE Spectrum,*

60, 28-33. Retrieved 26th February 2024. Available at: https://ieeexplore.ieee.org/document/10061631

Bogachev, R., & Zarikovskaya, N. (2021). Overview of the capabilities of the FastAPI web framework for implementing the backend of web systems. *Proceedings of the All-Russian Scientific Conference on Achievements of Science and Technology (DNiT-2021)*. Retrieved 8th April 2024. Available at: http://ru-conf.domnit.ru/media/filer_public/e0/09/e0093ef8-c0a2-48c7-a759-5ac0cc42d5ab/3015-dnit-2021.pdf

Bonney, M.S., de Angelis, M., Dal Borgo, M., Andrade, L., Beregi, S., Jamia, N., & Wagg, D.J. (2022). Development of a digital twin operational platform using Python Flask. *Data-Centric Engineering, 3*. Retrieved 8th April 2024. Available at: https://www.cambridge.org/core/journals/data-centric-engineering/article/development-of-a-digital-twin-operational-platform-using-python-flask/C138956055FE433E7EEB693777B0F339

Bozorovich, S.K. (2021). The Importance of Learning Culture Within a Language. Retrieved January 29th, 2024. Available at https://www.semanticscholar.org/paper/THE-IMPORTANCE-OF-LEARNING-CULTURE-WITHIN-A-Bozorovich/f4464a37ca262f7ce309683f389b8d29849f8985

Buchanan, B.G. (2005). A (Very) Brief History of Artificial Intelligence. *AI Mag., 26*, 53-60. Retrieved on 29th February 2024. Available at: https://ojs.aaai.org/aimagazine/index.php/aimagazine/article/view/1848

Carnegie Mellon University. Word Embedding Demo: Tutorial. Retrieved on 27th of March 2024. Available at: https://www.cs.cmu.edu/~dst/WordEmbeddingDemo/tutorial.html

Chisega-Negrilă, A. (2023). The New Revolution in Language Learning: The Power of Artificial Intelligence and Education 4.0. Bulletin of "Carol I" National Defence University. Retrieved 30 January 2024. Available at https://pdfs.semanticscholar.org/24ce/a3acfe1f898b3d2d4bcaed769e5bab1d3b27.pdf

Community-Oriented Development. (2014). *Sustainability, 39*(6-22). Retrieved on 24th April 2024. Available at: https://www.semanticscholar.org/paper/fe944ba5be4eb52e177277749e57e78b79e9ff35

Depue, W. (2023). What's new with DALL·E 3? OpenAI Cookbooks. Retrieved on 14th of March 2024. Available at: https://cookbook.openai.com/articles/what_is_new_with_dalle_3

Dhillon, S.K., Ganggayah, M.D., Sinnadurai, S., Lio', P., & Taib, N.A. (2022). Theory and Practice of Integrating Machine Learning and Conventional Statistics in Medical Data Analysis. *Diagnostics, 12*. Retrieved on 1st of March 2024. Available at: https://www.semanticscholar.org/reader/9ae2b575e2967cfaffdfafbe183de0b802da9df0

Duolingo Team (2023). Introducing Duolingo Max, a learning experience powered by GPT-4. Retrieved on 26th February 2024. Available at: https://blog.duolingo.com/duolingo-max/

Edmonds, E.A., Reid, I., Heggie, S., & Cornali, D.J. (1992). Constructing front-ends to existing software systems. Retrieved on 25th of March 2024. Available at: https://link.springer.com/chapter/10.1007/978-1-4471-3548-7_8

Eland, M. The Role of Web APIs in Modern Software Development. Retrieved on 8th April 2024. Available at: https://www.techelevator.com/the-role-of-web-apis-in-modern-software-development/

ElevenLabs API. Text to Speech. Retrieved on 3rd of May 2024. Available at: https://elevenlabs.io/docs/api-reference/text-to-speech

ElevenLabs. About ElevenLabs. Retrieved on 3rd of May 2024. Available at: https://elevenlabs.io/about

Emery, C (2022). Babbel Review: A Look At Price, Languages, and Alternatives. Retrieved on 28th February 2024. Available at: https://www.langoly.com/babbel-review/

Flutter. Flutter on Mobile. Retrieved on 27th of March 2024. Available at: https://flutter.dev/multi-platform/mobile

Frozzini, J. (2021). Interaction. *Anthropen*. Retrieved on 28th February 2024. Available at: https://doi.org/10.47854/anthropen.vi0.51159

Gąsior, K. (2019). Classroom Interaction in Oral Language Learning Events Created by EFL Storytelling-based Games. *New Horizons in English Studies*. Retrieved 1 February 2024. Available at : https://www.semanticscholar.org/reader/9b698c8eaed8b1dd99579e8b2cd46f11e95260e3

Gillis, A. natural language processing (NLP). *TechTarget, Enterprise AI*. Retrieved on 7th of March 2024. Available at: https://www.techtarget.com/searchenterpriseai/definition/natural-language-processing-NLP

Goodrich J. (2021). How IBM's Deep Blue Beat World Champion Chess Player Garry Kasparov. Retrieved on 29th February 2024. Available at: https://spectrum.ieee.org/how-ibms-deep-blue-beat-world-champion-chess-player-garry-kasparov

Gordon, S.R., Tarafdar, M., Cook, R.E., Maksimoski, R., & Rogowitz, B.E. (2008). Improving the Front End of Innovation with Information Technology. *Research-Technology Management, 51*, 50 - 58. Retrieved on 25th of March 2024. Available at: https://www.tandfonline.com/doi/abs/10.1080/08956308.2008.11657505

Górriz, Juan Manuel et al. Artificial intelligence within the interplay between natural and artificial computation: Advances in data science, trends and applications. *Neurocomputing* 410 (2020): 237-270. Retrieved on 29th February 2024. Available at: https://doi.org/10.1016/j.neucom.2020.05.078

Greyling, C. (2022). The Large Language Model Landscape**.** Retrieved on 7th of March 2024. Available at: https://cobusgreyling.medium.com/the-large-language-model-landscape-9da7ee17710b

Guetari, R., Ayari, H. & Sakly, H. Computer-aided diagnosis systems: a comparative study of classical machine learning versus deep learning-based approaches. Knowl Inf Syst 65, 3881–3921 (2023). Retrieved on 7th of March 2024. Available at: https://doi.org/10.1007/s10115-023-01894-7

Hansman, C.A. (2001). Context‐Based Adult Learning. New Directions for Adult and Continuing Education, 2001, 43-52. Retrieved 1 February 2024. Available at: https://onlinelibrary.wiley.com/doi/10.1002/ace.7

Harry, A. (2023). Role of AI in Education. Interdisciplinary Journal and Humanity (INJURITY). Retrieved January 30, 2024. Available at https://injurity.pusatpublikasi.id/index.php/inj/article/view/52

IBM. a. (2023). Understanding the different types of artificial intelligence. Retrieved on 1st of March 2024. Available at: https://www.ibm.com/blog/understanding-the-different-types-of-artificial-intelligence/

IBM. b. What is machine learning? Retrieved on 1st of March 2024. Available at: https://www.ibm.com/topics/machine-learning

IBM. c. What is overfitting? Retrieved on 5th of March 2024. Available at: https://www.ibm.com/topics/overfitting

IBM. d. What is underfitting? Retrieved on 5th of March 2024. Available at: https://www.ibm.com/topics/underfitting

IBM. e. What is a neural network? Retrieved on 5th of March 2024. Available at: https://www.ibm.com/topics/neural-networks

Islam, R. (2011). PHP and MongoDB Web Development Beginner's Guide. Retrieved 8th April 2024. Available at: https://www.semanticscholar.org/paper/PHP-and-MongoDB-Web-Development-Beginner's-Guide-Islam/32c155e002c419ee6b961b8bd8b0bb34ca3adc68

ITPro Today. (2022). Why APIs Are the Foundation of Modern Software Development. Retrieved on 8th April 2024. Available at: https://www.itprotoday.com/development-techniques-and-management/why-apis-are-foundation-modern-software-development

Jaleniauskiene, E., Lisaitė, D., & Daniusevičiūtė-Brazaitė, L. (2023). Artificial Intelligence in Language Education: A Bibliometric Analysis. *Sustainable Multilingualism, 23*, 159 - 194. Retrieved on 26th February 2024. Available at: https://intapi.sciendo.com/pdf/10.2478/sm-2023-0017

JavaTPoint. a. Turing Test in AI. Retrieved on 29th February 2024. Available at: https://www.javatpoint.com/turing-test-in-ai

JavaTPoint. b. React Features. Retrieved on 29th February 2024. Available at: https://www.javatpoint.com/react-features

Jayabharathi, S., & Ilango, V. (2021). A Brief Revolution of Evolution and Resurgence on Machine Learning. *2021 Asian Conference on Innovation in Technology (ASIANCON)*, 1-5. Retrieved on 29th February 2024. Available at: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9544706

Jones, R.H., Garralda, A., Li, D.C., & Lock, G. (2006). Interactional dynamics in on-line and face-to-face peer-tutoring sessions for second language writers. *Journal of Second Language Writing, 15*, 1-23. Retrieved on 28th February 2024. Available at: https://www.sciencedirect.com/science/article/abs/pii/S1060374305000561?via%3Dihub

Kamali K. (2023). The Revolution of Deep Learning in 2012: A Paradigm Shift in Artificial Intelligence. Retrieved on 29th February 2024. Available at: https://medium.com/@kaveh.kamali/the-revolution-of-deep-learning-in-2012-a-paradigm-shift-in-artificial-intelligence-d4fdbfa87a42

Kanade, V. (2023). What Is a Server? Definition, Types, and Features. Retrieved on 1st of April 2024. Available at: https://www.spiceworks.com/tech/tech-general/articles/what-is-a-server/amp/

Khan, F.H., Pasha, M.A., & Masud, S. (2021). Advancements in Microprocessor Architecture for Ubiquitous AI—An Overview on History, Evolution, and Upcoming Challenges in AI Implementation. *Micromachines, 12*. Retrieved on 29th February 2024. Available at: https://www.semanticscholar.org/reader/8989538ac21c85ec44f5f5f44ecf3a678e983682

Kim, M., Xin, Q., Sinha, S., & Orso, A. (2022). Automated test generation for REST APIs: no time to rest yet. *Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis*. Retrieved 8th April 2024. Available at: https://dl.acm.org/doi/10.1145/3533767.3534401

Kioko, A.N. (2018). The Importance of Foreign Language in Education. Chemchemi International Journal of Humanities and Social Sciences. Retrieved January 29, 2024. Available at https://journal.ku.ac.ke/index.php/chemchemi/article/view/1

Klassner, F. (1996). Artificial intelligence: introduction. Retrieved on 29th February 2024. Available at: https://dl.acm.org/doi/10.1145/332148.332149

Kollmannsberger, S., D'Angella, D., Jokeit, M., & Herrmann, L.A. (2021). Fundamental Concepts of Machine Learning. *Deep Learning in Computational Mechanics*. Retrieved on 1st of March 2024. Available at: https://link.springer.com/chapter/10.1007/978-3-030-76587-3_2

Kong, D., Chen, Q., Bao, L., Sun, C., Xia, X., & Li, S. (2022). *Recommending Code Reviewers for Proprietary Software Projects: A Large Scale Study*. Retrieved 24th April 2024. Available at: https://www.semanticscholar.org/paper/7fed76482e72fbe76461379408af98f517b075a9

Krishna, C.V., Rohit, H.R., & Mohana (2018). A Review of Artificial Intelligence Methods for Data Science and Data Analytics: Applications and Research Challenges. *2018 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2018 2nd International Conference on*, 591-594. Retrieved on 29th February 2024. Available at: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8653670

Lampou, R. (2023). The Integration of Artificial Intelligence in Education: Opportunities and Challenges. Review of Artificial Intelligence in Education. Retrieved January 29, 2024. Available at https://educationai-review.org/revista/article/view/15

Lathkar, M. (2023). High-Performance Web Apps with FastAPI: The Asynchronous Web Framework Based on Modern Python. *High-Performance Web Apps with FastAPI*. Re-

trieved 8th April 2024. Available at: https://link.springer.com/book/10.1007/978-1-4842-9178-8

Lin, C.H., Zheng, B., & Zhang, Y. (2017). Interactions and learning outcomes in online language courses. *Br. J. Educ. Technol., 48*, 730-748. Retrieved on 28th February 2024. Available at: https://bera-journals.onlinelibrary.wiley.com/doi/10.1111/bjet.12457

Linares-Pellicer J., Izquierdo-Doménech J., & Ferri-Molla I (2023). The new revolution of Generative AI models. GitBook. Retrieved on 29th February 2024.

Mendes, A. & Rodrigues, O. (2024). Top 10 best front end frameworks in 2024. Retrieved on 27th of March 2024. Available at: https://www.imaginarycloud.com/blog/best-frontend-frameworks/

Mondly. How does Luna use generative AI? Retrieved on 28th February 2024. Available at: https://www.mondly.com/faq/luna

Morandín-Ahuerma, Fabio (2022). ¿What is Artificial Intelligence? Int. J. Res. Publ. Rev 3 (12):1947-1951. Retrieved January 29, 2024. Available at https://philarchive.org/rec/MORQEI-2

Munday, P. (2016). The case for using DUOLINGO as part of the language classroom experience Duolingo como parte del curriculum de las clases de lengua extranjera. Retrieved 26th February 2024. Available at: https://www.semanticscholar.org/reader/0866d20c1fd590507ac7f15a867729bb6853ceee

NestJS. *NestJS - A progressive Node.js framework*. Retrieved 8th April 2024. Available at: https://nestjs.com/

Newborn, M. (2000). Deep Blue's contribution to AI. *Annals of Mathematics and Artificial Intelligence, 28*, 27-30. Retrieved on 29th February 2024. Available at: https://link.springer.com/article/10.1023/A:1018939819265

Noble Desktop. (2024). Top 5 Programming Languages Front End Web Developers Should Know. Retrieved on 27th of March 2024. Available at: https://www.nobledesktop.com/classes-near-me/blog/top-programming-languages-for-front-end-developers

Okal, B. O. (2014). Benefits of multilingualism in education. Retrieved on 30th November 2023. Available at https://repository.maseno.ac.ke/handle/123456789/3075

Okdie, B.M., Guadagno, R.E., Bernieri, F.J., Geers, A.L., & Mclarney-Vesotski, A.R. (2011). Getting to know you: Face-to-face versus online interactions. *Comput. Hum. Be-*

hav., 27, 153-159. Retrieved on 28th February 2024. Available at: https://www.sciencedirect.com/science/article/abs/pii/S074756321000213X?via%3Dihub

OpenAI API a. Vision Model. Retrieved on 3rd of May 2024. Available at: https://platform.openai.com/docs/guides/vision

OpenAI API b. Whisper. Retrieved on 3rd of May 2024. Available at: https://platform.openai.com/docs/guides/speech-to-text

OpenAI. (2023). Research GPT-4. Retrieved on 14th of March 2024. Available at: https://openai.com/research/gpt-4

Panesar, A. (2019). What Is Machine Learning? *Machine Learning and AI for Healthcare*. Retrieved on 1st of March 2024. Available at: https://doi.org/10.1007/978-1-4842-3799-1_3

Ping, Q., Niu, F., Thattai, G., Chengottusseriyil, J., Gao, Q., Reganti, A.N., Rajagopal, P., Tur, G., Hakkani-Tür, D.Z., & Nataraja, P. (2020). Interactive Teaching for Conversational AI. *ArXiv, abs/2012.00958*. Retrieved on 26th February 2024. Available at: https://www.semanticscholar.org/reader/074f286f422bd6205bef3397199fba7112f03747

Posey, B. What is a Server? Retrieved on 1st of April 2024. Available at: https://www.techtarget.com/whatis/definition/server

Putistina, O.V. (2015). Interaction in the Concept of Autonomous Language Learning. *Humanities and social sciences, 8*, 1919-1925. Retrieved 1 February 2024. Available at:https://elib.sfukras.ru/bitstream/handle/2311/19803/14_Putistina.pdf;jsessionid=EACBA9DBC7C41512BD0318F2F60E6D89?sequence=1

Ramírez, J., & Ortíz, A. (2020). A Review on Artificial Intelligence for Smart Computing Systems. Automation and Autonomous System, 12, 83-86. Retrieved January 29, 2024. Available at https://www.semanticscholar.org/paper/A-Review-on-Artificial-Intelligence-for-Smart-Ram%C3%ADrez-Ort%C3%ADz/e228c4e786316f5fd1452947d6c2fb58bf9eecc2

Rebolledo Font de la Vall, R., & González Araya, F. (2023). Exploring the Benefits and Challenges of AI-Language Learning Tools. *International Journal of Social Sciences and Humanities Invention*. Retrieved 31 January 2024. Available at https://valleyinternational.net/index.php/theijsshi/article/view/3771

Russell, S.J., & Bohannon, J. (2015). Artificial intelligence. Fears of an AI pioneer. *Science, 349 6245*, 252. Retrieved on 29th February 2024. Available at: https://pubmed.ncbi.nlm.nih.gov/26185241/

Sakalauskė, A., & Leonavičiūtė, V. (2022). STRATEGIC ANALYSIS OF DUOLINGO LANGUAGE LEARNING PLATFORM. *Mokslas - Lietuvos ateitis*. Retrieved on 26th February 2024. Available at: https://journals.vilniustech.lt/index.php/MLA/article/view/17731

Santa María Morales, S. (2024). The Era of Generative AI: How Large Multimodal Models are Reshaping Industries in 2024. Retrieved on 7th of March 2024. Available at: https://www.linkedin.com/pulse/year-large-multimodal-model-lmm-santiago-santa-mar%C3%ADa-morales-r1sse/?trk=articles_directory

Savvani, S. (2018). State-of-the-Art Duolingo Features and Applications. *International Conference on Interactive Collaborative Learning*. Retrieved on 26th February 2024. Available at: https://repository.essex.ac.uk/25974/1/state_of_the_art_duolingo_features_and_applications.pdf

Schneider, J. (2022). Foundation models in brief: A historical, socio-technical focus. *ArXiv, abs/2212.08967*. Retrieved on 6th of March 2024. Available at: https://www.semanticscholar.org/reader/775b2dc88cf04993f8596332444a906bec2db807

Sha, G. (2009). AI-based chatterbots and spoken English teaching: a critical analysis. *Computer Assisted Language Learning, 22*, 269 - 281. Retrieved 30 January 2024. Available at https://www.tandfonline.com/doi/abs/10.1080/09588220902920284

Shahid, S. (2023). Top Programming Languages in 2023: Insights from the Stack Overflow Developer Survey. Retrieved on 27th of March 2024. Available at: https://medium.com/@shafaggillani8/top-programming-languages-in-2023-insights-from-the-stack-overflow-developer-survey-8bbb8e490a15

Shaveta (2023). A review on machine learning. *International Journal of Science and Research Archive*. Retrieved on 1st of March 2024. Available at: https://pdfs.semanticscholar.org/f18c/08c5990b3d62f1dc85782d089a824309c89b.pdf

Shin, Y. (2019). The Spring of Artificial Intelligence in Its Global Winter. *IEEE Annals of the History of Computing, 41*, 71-82. Retrieved on 29th February 2024. Available at: https://ieeexplore.ieee.org/document/8736780/

Shortt, M., Tilak, S., Kuznetcova, I., Martens, B., & Akinkuolie, B. (2021). Gamification in mobile-assisted language learning: a systematic review of Duolingo literature from public release of 2012 to early 2020. *Computer Assisted Language Learning, 36*, 517 - 554. Retrieved 26th February 2024. Available at: https://www.tandfonline.com/doi/pdf/10.1080/09588221.2021.1933540?needAccess=true

Simmons, L. (2023). Front-End vs. Back-End: What's the Difference? Retrieved on 25th of March 2024. Available at: https://www.computerscience.org/bootcamps/resources/frontend-vs-backend/

Singapore Computer Society (2023). Artificial Intelligence (AI) in Singapore: Examples, Benefits, Courses And Opportunities. Retrieved on 29th February 2024. Available at: https://www.scs.org.sg/articles/ai-singapore

Son, J., Ružić, N.K., & Philpott, A. (2023). Artificial intelligence technologies and applications for language learning and teaching. *Journal of China Computer-Assisted Language Learning, 0*. Retrieved 31 January 2024. Available at https://www.degruyter.com/document/doi/10.1515/jccall-2023-0015/html

TalkPal AI. Babbel vs Rosetta Stone: Which Language Learning App Is More Efficient? Retrieved on 28th February 2024. Available at: https://talkpal.ai/babbel-vs-rosetta-stone/

Turing. A Guide on Word Embeddings in NLP. Retrieved on 7th of March 2024. Available at: https://www.turing.com/kb/guide-on-word-embeddings-in-nlp

Warahmah, M. (2022). The Importance of Language Learning in the Curriculum of 2013. Formosa Journal of Multidisciplinary Research. Retrieved January 29, 2024. Available at https://www.semanticscholar.org/paper/The-Important-of-Language-Learning-in-the-of-2013-Warahmah/5f9f920f08d3c3eb2167cebce60246391ee083f3

Wu, A., Caddell, J., & MacNeil, C. (1986). Introduction to Artificial Intelligence. Retrieved on 29th February 2024. Available at: https://arc.aiaa.org/doi/10.2514/6.1986-163

Yalçın, O.G. (2021). Introduction to Machine Learning. In: Applied Neural Networks with TensorFlow 2. Apress, Berkeley, CA. Retrieved on 29th February 2024. Available at: https://doi.org/10.1007/978-1-4842-6513-0_2

Yin, H., & Shi, L. (2021). Which Type of Interpersonal Interaction Better Facilitates College Student Learning and Development in China: Face-to-Face or Online? *ECNU Review of Education*, 209653112110108. Retrieved on 28th February 2024. Available at: https://journals.sagepub.com/doi/10.1177/20965311211010818

Yogasudha, S., Mounika, K.V., Namitha, P.R., & Priya, K.M. (2021). Deep Learning. *Machine Learning — A Journey to Deep Learning*. Retrieved on 5th of March 2024. Available at: https://doi.org/10.1142/9789811234064_0012

Zaman, B.U. (2023). Transforming Education Through AI, Benefits, Risks, and Ethical Considerations. Retrieved January 30, 2024. Available at https://vixra.org/pdf/2310.0047v1.pdf

Zhang, M., Belhadi, A., & Arcuri, A. (2023). JavaScript SBST Heuristics to Enable Effective Fuzzing of NodeJS Web APIs. *ACM Transactions on Software Engineering and Methodology, 32*, 1 - 29. Retrieved 8th April 2024. Available at: https://dl.acm.org/doi/10.1145/3593801

ZhangMan, & ArcuriAndrea (2022). Adaptive Hypermutation for Search-Based System Test Generation: A Study on REST APIs with EvoMaster. *ACM Transactions on Software Engineering and Methodology, 31*, 1-52. Retrieved 8th April 2024. Available at: https://dl.acm.org/doi/10.1145/3464940

Zhu, A. (2020). Application of AI Identification Technology in Foreign Language Education. *2020 International Conference on Artificial Intelligence and Education (ICAIE)*, 71-75. Retrieved on 26th February 2024. Available at: https://ieeexplore.ieee.org/document/9262550