Abstract

Author(s)	Publication type	Completion year
Aliaga Torró, Carlos	Thesis, UAS	2023
	Number of pages 61	

Title of the thesis

AI-based educational learning application

Degree, Field of Study

Bachelor of Engineering, Information and Communication Technology

Organisation of the client

Abstract

Explores the potential of artificial intelligence in education by creating an application that transcribes videos and answers questions related to the videos using open-source models. The AI-based platform allows educators to upload online classes or educational videos, which are automatically transcribed. Users can also ask questions related to the transcriptions and receive answers with the timestamp where the explanation is given. The system also offers an application programming interface (API) to make it easier for educational institutions to implement the application. The thesis aims to demonstrate how artificial intelligence can enhance the quality of education and reduce costs by utilizing open-source technology.

Resumen

Esta tesis explora el potencial de la inteligencia artificial en la educación mediante la creación de una aplicación que transcribe vídeos y responde a preguntas relacionadas con los vídeos utilizando modelos de código abierto. La plataforma basada en IA permite a los educadores subir clases en línea o vídeos educativos, que luego se transcriben automáticamente. Los usuarios también pueden hacer preguntas relacionadas con las transcripciones y recibir respuestas con la marca de tiempo en la que se da la explicación. Además, el sistema ofrece una interfaz de programación de aplicaciones (API) para facilitar a los centros educativos la implementación de la aplicación. La tesis pretende demostrar cómo la inteligencia artificial puede mejorar la calidad de la educación y reducir costes utilizando tecnología de código abierto.

Keywords

Artificial Intelligence, education, natural language processing, transcription, question-answer, API

Palabras Clave

Inteligencia Artificial; Educación; Procesamiento de Lenguaje Natural; Transcripción; Pregunta-Respuesta; API

Table of Contents

List of abbreviations/concepts/terms

AI: artificial intelligence

API: application programming interface

Decoder Layer: component of the transformer architecture generates the output data based on the input data and the keys and values generated by the encoder layers.

Encoder Layer: component of the transformer architecture processes the input data and extracts relevant information for a given task.

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

GUI: graphical user interface

HCI: Human-Computer Interaction

LLM: Large Language Model

MLM: Masked-Language Model

NLP: Natural Language Processing

Script: programs or sequences of instructions that are interpreted and used to automatise tasks

UI: user interface

1 Introduction

In recent years, the field of artificial intelligence has been advancing rapidly. However, the high cost of resources, including data, hardware, and energy, makes it challenging to build for low-resource companies. As a result, the potential benefits of AI for society may not be fully used. Nevertheless, larger companies publish open-source models that can provide small developers with the tools they need to create innovative applications. Using those models reduces the economic and environmental costs of training them as they consume vast amounts of energy. In the context of this thesis, the objective is to explore the use of AI for educational purposes and the power of open-source models in achieving this goal. In more detail, the thesis aims to examine how AI can improve education and how open-source models can be used to achieve the objective.

The theoretical section will delve into the role of AI in transforming education and its various applications. It will provide an overview of the history of AI, its different types, and how they have evolved, with a specific focus on machine learning and deep learning. The section will also explore natural language processing, transformers, and software used in AI development. The theoretical section will also touch on the basics of frontend and backend development, including user interfaces, mock-ups, and prototypes. It will then provide a more detailed examination of backend development, discussing servers, APIs, message brokers, task queues, languages, and database design.

The practical aspect of this study focuses on developing a REST API designed explicitly for the learning environment. The goal is to create a system that allows teachers to upload various video materials or online class recordings and for students to ask related questions. When a student has a question, it can be asked to the system and answered with the information extracted from the video. The application will indicate the segment where the answer is quoted. Moreover, the sources from which the information has been obtained will also be indicated. This feature allows students to review the content and clarify any doubts. A web application is developed to demonstrate its usage case and potential. The API development enables different institutions to apply the system on their interface while maintaining design guidelines.

2 Education

2.1 Introduction

Education is a process that aims to the development knowledge, habits and skills required to live in society. It prepares individuals for life by training and guiding them with the necessary knowledge. Several institutions work in order to acquire essential education for society. The individual's education is a lifelong process that keeps developing cognitive, social, emotional and physical capabilities. Education is considered *a human right, a public good, and a critical driver for sustainable development* (Locatelli 2018). (Sharna 2022.)

Social development is a consequence of education leading to the acquisition of social and emotional skills. Although education is a lifetime process, it mainly occurs in childhood and adolescence, allowing one to establish stable relationships with family, friends and others. Social development correlates with education and people's capacity to interact with other environments, so education is an important part of the development of social skills. Respect and communication skills are some capabilities that are developed during this process. (CIS 2019.)

Education is also considered a factor in the development of a community. Education helps to break barriers and promote tolerance and cooperation. Community development and the education of the people that integrate it improve the quality of life, making it more sustainable and prosperous. Through education, individuals learn how to make the right decisions and use their assets to influence a decision that influences the community. The collaborative approach of sharing experiences can also be taken as learning for other individuals. (Billion Acts.)

In terms of skills, people need to be able to communicate, deal with conflicts and learn how to transform them constructively. Regarding attitudes and values, people must keep motivation and commitment to protecting human dignity, empathy and solidarity for order. There must be a sense of justice and responsibility for the own actions. People should feel confident and able to address and combat gender inequalities and gender stereotypes. Education provides a framework to address and deal with these differences in understanding values. (Council of Europe portal 2015.)

However, there is a difference between education and knowledge. The process of acquiring, processing, understanding, and recalling information through one of several methods is known as knowledge. Knowledge acquisition details how people experience new information and how it can be recalled later. When information is retrieved, a process of un-

derstanding is carried out to encode it so a person can build a cognitive model. (Wiesen 2023.)

2.2 Transforming education with AI

Artificial intelligence can be defined as a simulation of human intelligence in machines (Frankenfield 2022). It is a growing field, and many industries are incorporating this technology to improve their product. Smart cities, business intelligence, cyber security, and manufacturing are some of these industries. One of the promising industries is healthcare, where diagnosis, research, early detection and training are making professionals more efficient and improving everyone's life. (Achary 2019.)

Education can also take advantage of these advancements. AI can analyse large amounts of data from each student to create personalised learning paths. The system is adapted to the student's necessities and abilities. Instant feedback can be provided in order to modify teaching methods. Another point is the opportunities to access quality educational resources in areas with a gap. The last point is efficiency, where AI can process repetitive tasks and analyse data. (Gülen 2023.)

However, all that glitters is not gold. The potential to automate many jobs leads to increased efficiency that may result in job losses. Relying on AI tools and platforms creates a dependency on technology, which leads to a lack of human interaction. As AI rely on data, bias and discrimination are implicit in the data. Data must be taken to improve for the development, leading to privacy concerns. (Gülen 2023.)

2.3 AI applications in education

Artificial intelligence is a technology that can personalize the experience in different industries. Concerning education, it can offer convenient options for adapting students' learning process, helping teachers and improving learning groups. (Artiba 2021.)

Some new methods are appearing for learning objectives that involve AI. Smart content consists of guides, conferences, and educational videos about the student's knowledge. During the realisation of tests and answer questions, data is collected from students leading to a personalisation of the learning path to adapt it to the needs of the person. As shown in Figure 1, it helps students adapt the content and the speed of lessons to their needs and fulfil their objectives. (Civati 2021.)

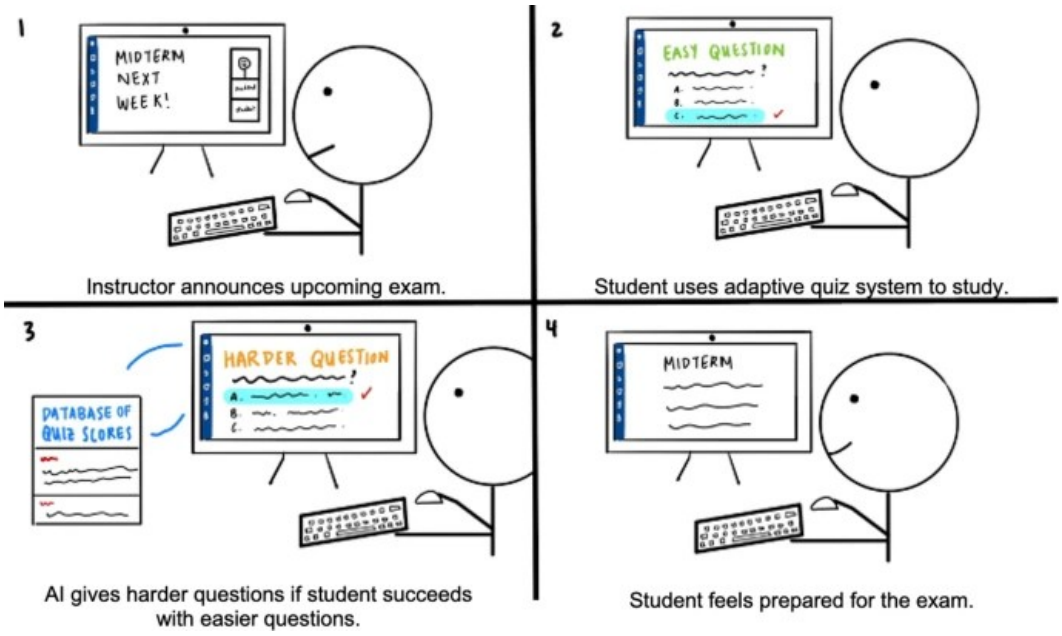

Figure 1. Storyboard example of Adaptive Quiz (Seo et al. 2021.)

Another tool is virtual learning environments that have started to develop thanks to virtual reality devices. The development of avatars in this environment provides more significant interaction between people in different locations. AI allows the internationalization of the content as it can translate, transcribe audio, and emulate voice in different languages. (Seo et al. 2021.)

Like other industries, education also changes the role that workers develop. In this case, teachers have to change their roles. As AI take functions such as grading test, real-world tutoring and choosing content, teachers could provide human interaction, helping students and providing hands-on experience. (Civati 2021.)

Promoting equity and accessibility in education is essential to reduce the gap between different communities. AI's potential is being explored to create tools for creating more accessible learning environments. Some technologies based on machine learning models, such as advanced speech synthesis or transcription of audio, allow for improving the user experience for students with disabilities as the time of producing new material adapted is reduced. In case students have blind or have low vision, AI can create a description of visual content. There are tools like voice assistants to read content from a page, summarize it, or skip content that is unimportant. (Educause 2022.)

2.4 State of art in AI for education

As can be seen, in the past few years, we had the problem of Covid-19 and lockdowns. Online and digital learning step up to fill the gap caused by the pandemic. AI has played a critical role in facilitating this process. Code developers have been developing some education-related applications that make life easier. Several examples use artificial intelligence to improve their product or as a base for the product.

Thinkster - https://hellothinkster.com/online-math-tutor.html

A Thinkster online math tutor can help a child develop the discipline, rigour and grit to become a math champion! (Thinkster). It introduces artificial intelligence to create personalized learning programs.

Cognii - https://www.cognii.com/

Cognii is a leading provider of Artificial Intelligence based educational technologies. They work with organizations in the K-12, higher education, and corporate training markets to help them deliver 21st-century online education with superior learning outcomes and cost efficiency. EdTech product is helping students worldwide by enabling personalized deeper learning, intelligent tutoring, open response assessments, and pedagogically rich analytics. (Cogni.)

Duolingo - https://www.duolingo.com/

Duolingo is a popular app for learning languages. The company's objective is to develop the best education in the world and make it universally available. Duolingo combines human expertise with smart AI to use the strength of both. As shown in Figure 2, the repetitive tasks and those that use a huge amount of data are done by artificial intelligence. (Duolingo 2022; Pajak & Bicknell 2022.)

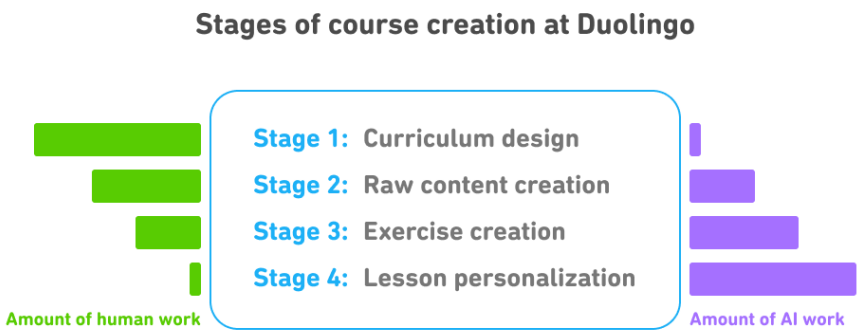

Figure 2. Stages of course creation human and AI (Pajak & Bicknell)

Gradescope - https://www.gradescope.com/

Gradescope is an AI tool that enables students to assess each other while providing feedback. It combines ML and IA to make the grading task easier. In this case, teachers can focus on other important tasks instead of grading. Some features are AI-assisted learning, student-specific time extensions, AI grading system, which accomplish increased efficiency and a fair system in grading. (McFarland 2022.)

3 Artificial Intelligence

3.1 Introduction

Artificial intelligence is the science and engineering of making intelligent machines, especially intelligent computer programs (McCarthy, 2007). However, this is not an easy task. There is an important concept that should be defined first, intelligence.

Intelligence is a process or an innate capacity to use information in order to respond to ever-changing requirements. It is the ability to solve problems by acquiring, adapting, modifying, extending and using information. Emotional and social intelligence is also included. Intelligence involves the capacity to adapt and learn accumulated through our species. The improvement by practice and study. (MacFarlane 2013.)

So according to this definition, Artificial Intelligence has the wrong name and should be defined as Artificial Knowledge (MacFarlane 2013). However, the AI development's main goal is to give computers the intelligence of humans. Artificial intelligence heavily depends on other fields, as Figure 3 shows. The process of data gathering and the work of machine learning engineers, data engineers, and data analysis, among others, is what allows the fast-paced development of this field.

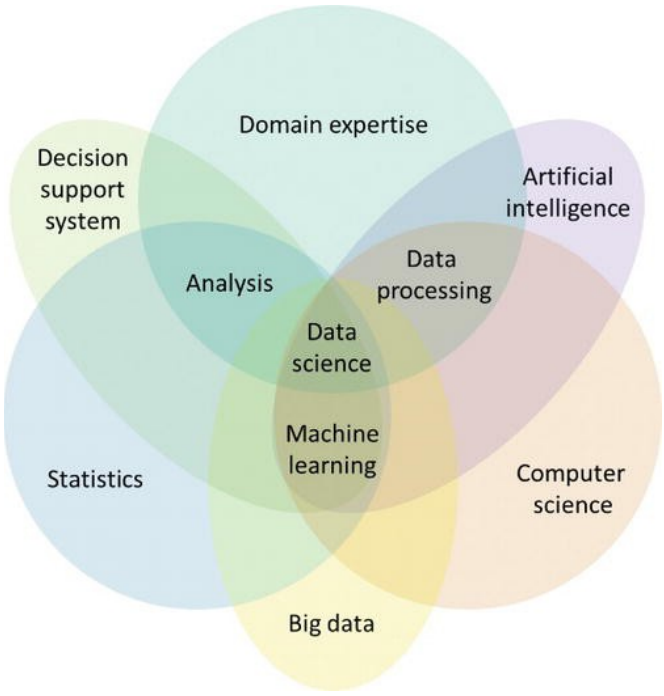

Figure 3. The field of data science including statistics,
big data, and artificial intelligence (Lee et al. 2019)

Defining AI is a complex task, encompassing many phenomena and concepts (Tilbe 2022). Historically, four approaches have been followed. As shown in Table 1, there are different definitions of AI based on the top of the table, thought or behaviour at the bottom. The ones on the left compare AI with human performance, whereas the right ones compare it with ideal performance.

Table 1. Definitions of AI depending of the author thought (Russell & Norvig 2009, 2.)

Thinking Humanly	**Thinking Rationally**
The exciting new effort to make computers think ... machines with minds, in the full and literal sense (Haugeland 1985). *[The automation of] activities that we associate with human thinking, activities such as decision-making, problem solving, learning ...* (Bellman 1978).	T*he study of mental faculties through the use of computational models* (Charniak & McDermott, 1985). *The study of the computations that make it possible to perceive, reason, and act* (Winston, 1992).
Acting Humanly	**Acting Rationally**
The art of creating machines that perform functions that require intelligence when performed by people (Kurzweil, 1990). *The study of how to make computers do things at which, at the moment, people are better* (Rich & Knight, 1991).	*Computational Intelligence is the study of the design of intelligent agents* (Poole et al., 1998). *AI . . . is concerned with intelligent behavior in artifacts* (Nilsson, 1998).

3.2 History

Everything started in 1950 when a generation of scientists, mathematicians and philosophers began to think about and understand the concepts surrounding artificial intelligence. They started to describe and conceptualise what we know today. Alan Turing started exploring the mathematical possibility of AI. In relation, he began to look for the pos-

sibility of replicating human thought on machines. He developed the Turing Test or also called the "Imitation Game". The first version involved a man, a woman and player C. They communicate by writing notes. The game's purpose is to determine which of the two is the man and which is the woman. So, the game is adapted to computers. A computer performs one role and the other by a man or a woman. The objective of this second version is to test when a computer behaves like a human. However, it took much work to accomplish the objective. Computers were costly and slow. (Turing 1950, 1; Anyoha 2017.)

The conference, considered the birthplace of IA, was held in 1956 and organised by John McCarty, Dartmouth Summer Research Project. It consisted of several weeks of mathematicians and scientists debating and brainstorming about IA. Nothing was achieved, but it conveyed the feeling that it was possible. "Development of Logic Theorist" was considered "the first artificial intelligence programme" and was presented at this convention. (Russell & Norvig 2021, 32.)

Even with all the attempts, artificial intelligence still failed to take off. During the next 30 years, the so-called "AI winter" as can be seen in Figure 4. It took place due to the lack of power in the computers of the time. Gradually, computers improved their power, speed, and storage and reduced their price considerably. (Russell & Norvig 2021, 42.)

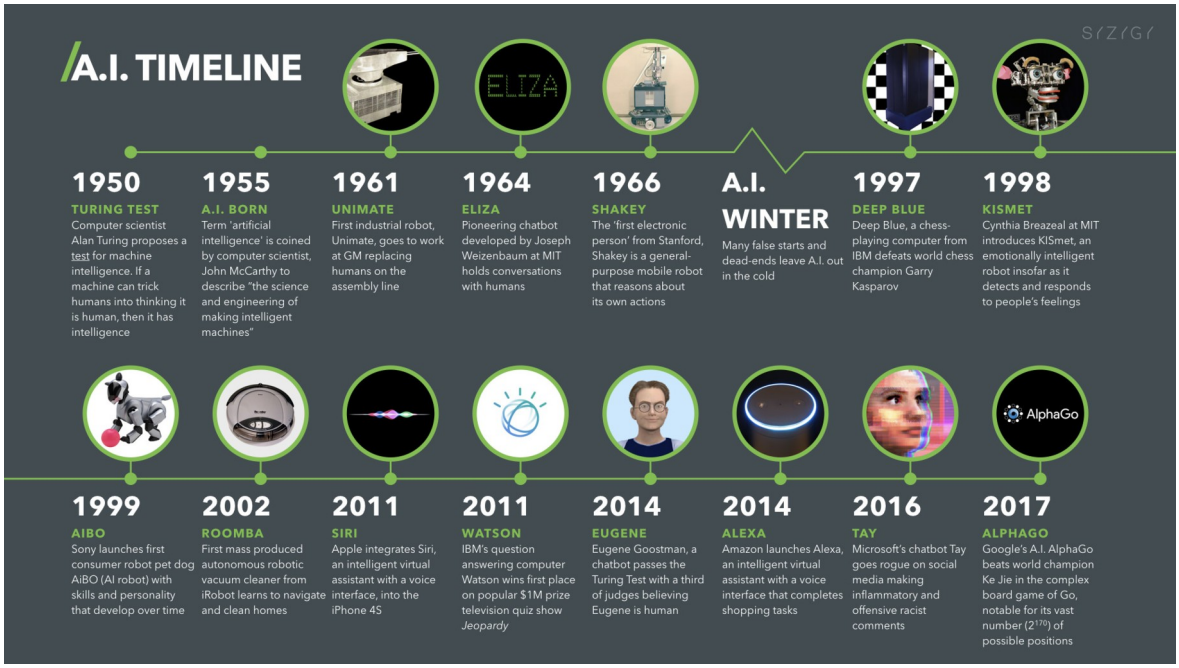

Figure 4. Artificial Intelligence Timeline (Marsden 2017)

The first honest attempt at the development was based on the technique of translating hu-

man knowledge onto the computer through a rule-based system. It is a rule-based system that combines human knowledge in a particular problem domain. The knowledge is represented in declarative form, and the system can operate with uncertain conditions or incomplete knowledge. CLIPS was the tool for building such systems. NASA developed it under the name "NASA's AI Language". The problem with this type of system was the hard manual work of writing down all the state rules to be considered. In addition, it has a limited capacity, as it is limited to the rules written down. (Doe 2020; Secret Society Software 2022.)

When computers had sufficient memory and processing speed, IBM Research developed a parallel computing computer called Deep Blue. This computer was able to play chess extensively. It was capable of exploring 200 million possibilities per second. The IBM computer in 1997 was pitted against the world champion and chess grandmaster, Garry Kasparov (IBM a). This event was a milestone in the development of AI as it won. Over the next few years, researchers continued to develop a programme capable of beating the champion Go game. This game is very complex as it has 10 to 170 possible configurations making it more complex than chess. The Alpha Go program developed by DeepMind (Google) was responsible for defeating the Chinese Go champion in 2015. The computer had 1920 CPUs and 280 GPUs (Hern 2016). It shows the amount of resources that are needed to train and run AI. (IBM b; DeepMind.)

3.3 Types of AI

Artificial intelligence can be classified depending on the type of limitations. We have two main branches, one related to the capability of the AI, based on the task and learning capability. The other branch takes into account the design features.

Capability-based AI:

- Narrow AI, also known as "weak AI". It is designed to perform a single task. However, it can outperform a human in a specific task. It is the type of AI that is currently being used. It is trained with a dataset with information about the specific topic, explaining why it is excellent in just one specific task. Due to the architecture and how it is trained, the type of AI is not conscious, sentimental or emotional. (Jajal 2018)
- General AI can do different tasks well, showing intelligence in a wide range of tasks. It can also outperform humans in different tasks. Regardless that they are machines, they are conscious, sentience and can think outside the box. It allows the machine to think of different strategies that can be taken to solve a problem

and think abstractly. As it is said, they can do what humans can do, including learning, planning tasks and solving problems, among other things. (Jajal 2018)

- Super AI is defined as *an intellect that is much smarter than the best human brains in practically every field, including scientific creativity, general wisdom and social skills* (Bostrom). This type of AI worries people because it will lead to the extinction of the human race as it will surpass human intelligence in every way. (Jajal 2018)

Design features AI:

- Reactive machines are the oldest form of AI systems that have limited capabilities. They can not learn or remember past experiences due to the lack of a memory-based system. They respond to a limited set of inputs, such as Deep Blue. (JavaT-Point)
- Limited memory systems have all capabilities of reactive machines, but they also can learn about historical data. It is the standard approach based on information in datasets. (JavaTPoint)
- Theory of mind includes the concept of understatement of what a machine is interacting with. It includes the ability to identify emotions and needs. It wants to achieve the human capability of understatement in multiple fields. (JavaTPoint)
- Self-awareness is a concept similar to the human brain, where we include needs, thoughts, emotions and desires. It is the goal of AI development that some authors describe as a double-edged sword. It could be chaotic and lead to the extinction of the human race, or it could be perfect for improving the human race. (JavaTPoint)

3.4 Machine Learning

Machine Learning is a subfield of artificial intelligence that uses different learning methods to simulate human behaviour and mentality. With these methods, AI can similarly solve complex tasks to humans would. Also, AI can make predictions using input data without being explicitly programmed, as shown in Figure 5. As a result, data science has become a growing field. (Brown 2021.)

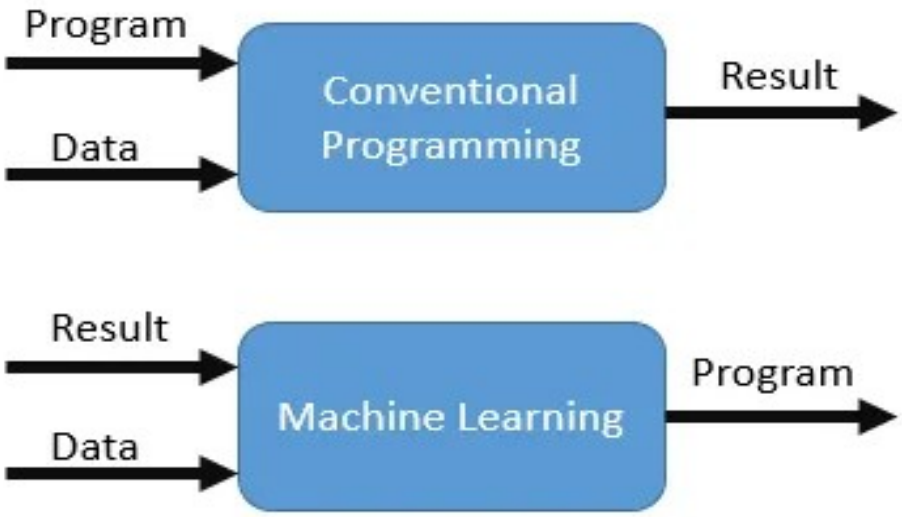

Figure 5. Conventional Programming vs Machine Learning (Stefanus 2019)

When AI is trained, we obtain a model. It accepts different parameters related to the input data used in the training process and returns an output. Arthur Samuel defines m*achine learning* as *the field of study that allows computers to learn without explicitly being programmed* (Samuel 1959). (Brown 2021.)

Approach

When training artificial intelligence, we have several strategies leading to different models. Each of them has different characteristics and requires a dataset. So, depending on our problem, we must adapt the strategy to the specific case.

The first strategy is supervised learning, a subset of machine learning that uses algorithms trained with labelled datasets. It consists of two parts, one is the data, and the other is the label that is used to classify the information. Supervised learning measures accuracy through a loss function which is a metric that can be used to measure how good our model is. We also have modified associated parameters to minimise the error and allow the algorithm to approach a good accuracy and make it "learn". Two types of problems can be solved with supervised learning. They are classified depending on the expected result. We will discuss a classification problem if we expect to classify data. However, we would face a regression problem if we expect a numerical value. (IBM c.)

The second strategy is unsupervised learning, which is based on the analysis and clustering of data. A dataset is provided, but in this case, it is unlabelled. It allows the AI to ex-

plore and analyse different sets of information, revealing hidden patterns that are difficult for a person to identify. (IBM d.)

The last strategy is reinforcement learning. It is based on algorithms that perform actions in an environment to maximise a pre-programmed reward. It does not require labelled data, as its rewards guide suitable actions, and if a lousy action is taken, it will be penalised. The algorithm will try to find a balance between exploration and current knowledge. (GeeksForGeeks 2023.)

As shown in Figure 6, each of the different approaches that can be taken has its characteristics. Depending on the nature of the problem and the data provided, different approaches can be taken, producing different outputs.

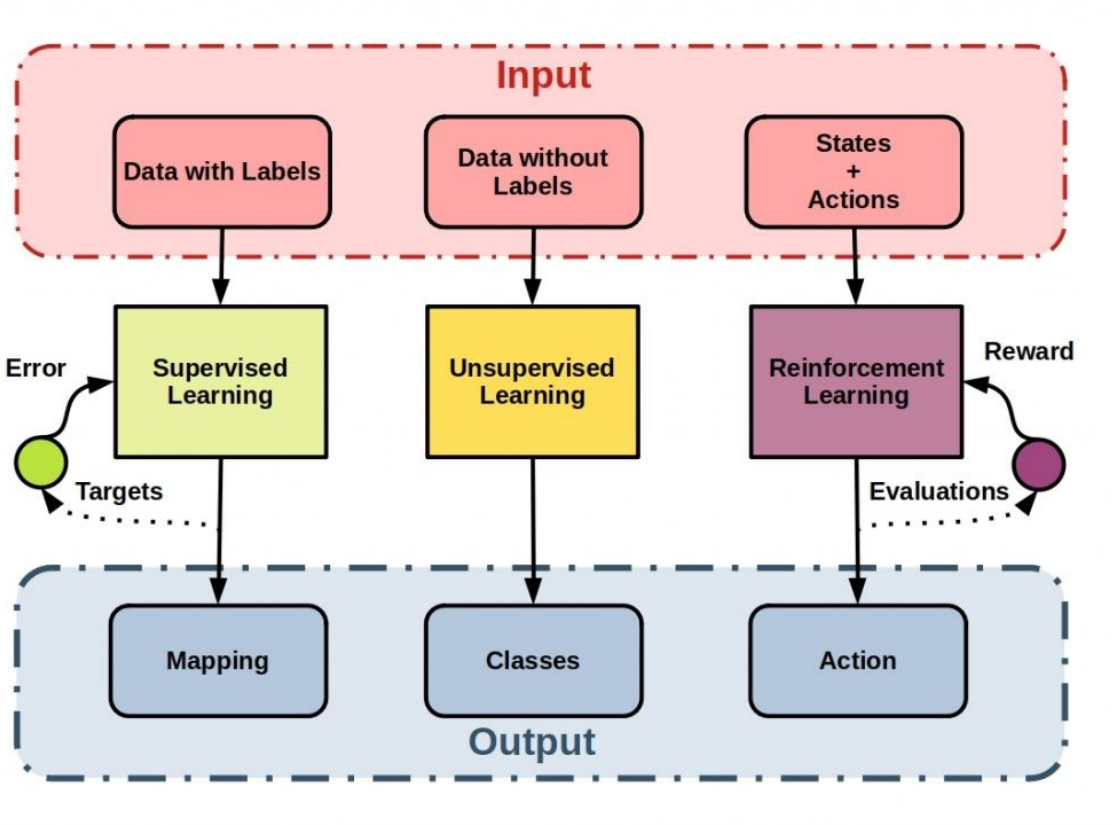

Figure 6. Supervised vs Unsupervised vs Reinforcement Learning (Rainergewalt 2021.)

Training problems

Training an artificial intelligence is *learning (determining) good values for all the weights and the bias from labelled examples* (Google 2022). This process is complex due to the

massive number of parameters involved. A simple parameter wrongly set has severe consequences in the trained model. One of the main problems we have when training our AI is the data, as machine learning is a big data consumer. It must be collected and stored in a specific format, ensuring quality. Some information can be mislabelled as having a direct impact on the performance of the AI model. (Wiggers 2021.)

Another problem is bias, an anomaly in the output related to the information the AI was trained. It may be due to incomplete data. There are also cognitive biases, which are unconscious human thinking errors affecting their judgements. They are found in datasets because they are introduced unknowingly, as over a hundred are defined in the human mind. (Dilmegani 2020.)

Finally, over-fitting is another major problem when a model has learned the information by heart. It cannot generalise with new information and generates an erroneous result, decreasing accuracy. There are several ways to avoid this problem. One is cross-validation, a technique based on splitting different parts of training data and using it to check the accuracy while we are training. (AWS a.)

Interpretability and explainability

In machine learning, models accept data and returns and output. However, explaining a result or reaching a decision is not given. Transparency is a characteristic that is not present in this technology. Even programmers do not know or understand how the output was reached. (Seon.)

Interpretability is the ability to define relationships between the input and output. In other words, the cause and effect can be determined. In case a decision is low-risk, the importance of interpretability loses weight. A highly interpretable model is desirable in a high-risk situation so the decision can be understood in case the user wants to explain how the decision was taken. (Johnson 2020.)

Explainability is a characteristic that explains what happens to the model from the input until the output is received. Explaining what features contribute to the model's prediction helps to understand machine learning models better. Three critical aspects of explainability are transparency, the ability to question, and ease of understanding. (Onose 2023.)

3.5 Deep learning

A neuron is a mathematical function of one or more inputs weighed relative to the weights of each input. Each value is summed, and an offset or bias is added, allowing slight modifications. (Dhingra 2021.)

AI neurons are designed to replicate aspects of biological neurons. Biological ones throw ions into neighbour neurons. It is simulated through the activation function and the output value of a neuron between -1 and 1. If the output of a neuron exceeds a threshold, it will send the value to the next layer of neurons. In Figure 7, we have the structure of the artificial intelligence model. (Nagyfi 2018.)

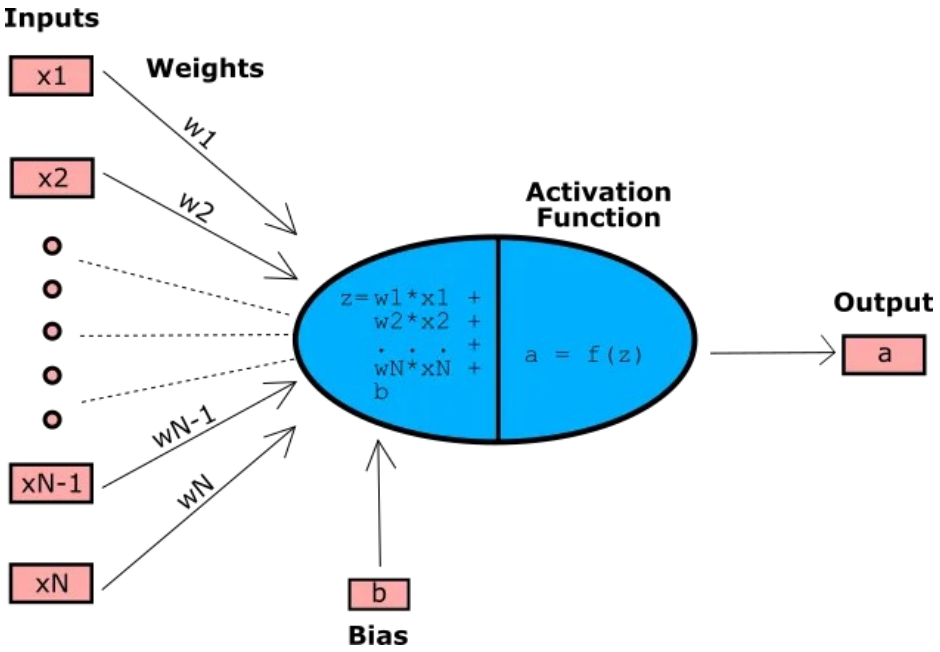

Figure 7. Structure of a neuron (Fulton 2018.)

Neural networks are the central part of deep learning algorithms. Based on a structure composed of interconnected layers, it tries to simulate the human brain, as seen in Figure 8. Each layer contains many neurons that depend on the type of layer. We can find three different layers. The first layer is the input layer containing n neurons equal to the number of input parameters provided. Secondly, there are several hidden layers where all the mathematical operations are made. The engineer who built the architecture decides the number of layers and the number of neurons on each layer. Finally, the number of neurons in the output layer depends on the problem to solve. (IBM e.)

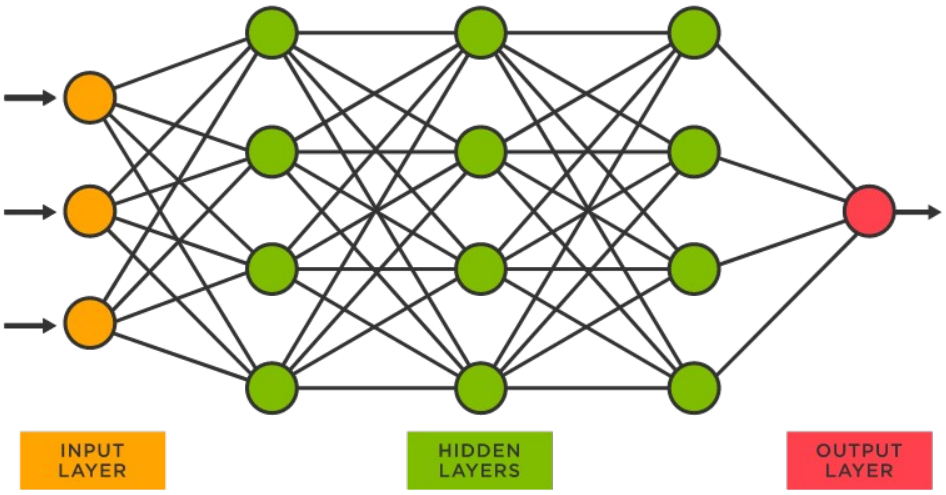

Figure 8. Structure of neural network (Tibco.)

3.6 Transformers

A transformer is a neural network architecture that learns context by tracking relationships in sequential data. It is mainly used for natural language text or audio signal processing, question answer, and information extraction. It was introduced first in Attention Is All You Need paper by Google. (Vaswani et al. 2017; Merritt 2022.)

Self-attention and attention are essential mechanisms that let transformers pay attention to the input and output when making predictions. It is a fundamental mechanism for tasks that must keep a relation between different words or sentences between input and output. Some of those tasks are translation, sentimental analysis and text synthesis. Codification-decodification architectures use attention. The encoder vectorizes the input sequence, and the decoder attends the encoder representation of the whole input to make a prediction. (Wydmanski 2022.)

Transformer layer architecture consists of several encoder and decoder layers composed of multiple self-attention and feed-forward layers. Encoder layers process the input data and generate a set of "keys" and "values" that capture the relevant information. The decoder layers use these keys and values along self-attention mechanisms to generate an output sequence. Transformers also included multi-headed self-attention, allowing the model to attend to multiple aspects of the input data simultaneously. It helps the model capture the natural language's complexity and improve performance. (Cristina, S. 2022.)

3.7 Natural Language Processing

A language can be defined as a set of rules or symbols combined and used to share information or broadcast it. Natural Language Processing (NLP) is a branch of Artificial Intelligence and Linguistics that tries to make computers understand the words written in human language. It is subdivided into natural language generation, which is the process of producing meaningful text. Another division is natural language understanding, defined as Linguistics, the science of language that includes phonology, syntax, morphology, semantics and pragmatics, as shown in Figure 9. So natural language processing focuses on allowing computers to understand human language. (Khurana et al. 2022.)

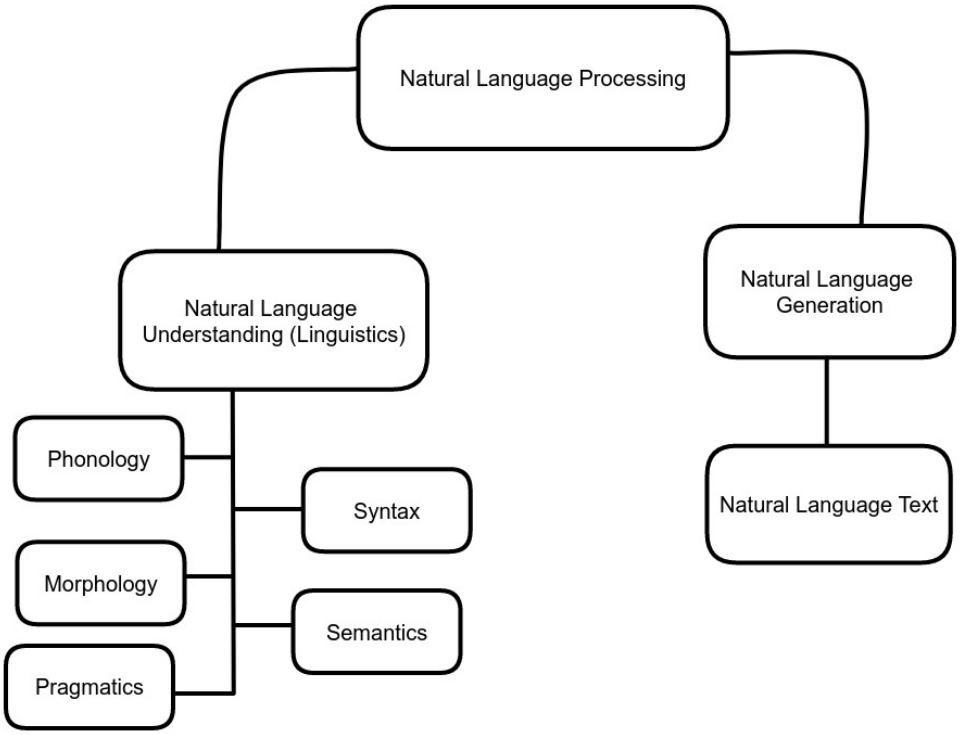

Figure 9. Clasification of NLP

NLP is growing fast due to the vast amount of generated data. Message apps, social media, forums, and blogs, among others, generate text that is unstructured and is not being used. NLP comes to help in the process of understanding the text handling large volumes of text data that the AI can process without fatigue and with consistency. (Lee 2019.)

Some of the most common techniques for natural language processing is the analysis and text classification. Named entity recognition is a technique that identifies and labels the

'named entities' inside a text and extracts them for post-analysis, as shown in Figure 10. Sentiment analysis follows a similar process to the one explained before to extract the mood of the sentence. Another application is text summarization, which makes summaries of complex text in basic terms easy to understand. Topic modelling is another technique of natural language processing no supervised that uses AI programs to group and label sets of text with a common topic. In the same way, it can classify massive amounts of no structured text for post-analysis. Finally, the last application is the extraction of keywords. It is an automatic process that extracts the most relevant information from a text using AI and machine learning algorithms. (Wolff 2021.)

Figure 10. Named Entity Recognition example (Wolff 2021.)

Natural Language Processing faces several challenges that limits its effectivity on real world application. The lack of context for homographs, homophones, and homonyms is a problem as words with similar sounds and pronunciation can be understand with different definition depending on the context. Ambiguity, errors on speed and text, slangs and colloquialisms are also different problems that this field has to challenge. (Ghiya 2022.)

Language Models and LLM

A language model is a statistical model that predicts the probability of a sequence of words in certain language. It is an important component of NLP systems required in several applications.

Two approaches to the language model can be taken. The first one is a probabilistic language model done by calculating n-gram probabilities. N-gram is a sequence of n words that occurs together in a text. The probability of n-gram is the chance that the last word in that group will appear based on the previous words in the group. The second approach is neural network-based language models. It has a neural network that can be seen as a function that returns a probability of the next word in a sentence, given an input. It uses word embedding to represent words as dimensional vectors and softmax functions to compute probability distributions over words in sentences. (Kapronczay 2022.)

A large language model is a type of language model that uses neural networks to generate natural text that is more coherent and can be used in a wide range of tasks compared to language models. It is because large language models have many parameters and are trained on much larger datasets than language models. They often use transformer architecture. These models usually are trained with unsupervised learning techniques. (Lee 2023.)

LLM are models trained with massive amounts of data that allow them to generate human-like language, more coherent and with mode deep knowledge compared to language models. ChatGPT is one of the fastest-growing and most known applications based on LLM. It is a sibling model to InstructGPT that, at the same time, is an improvement of GPT-3. It can generate, summarise, parse, classification, and translate text. (Dilmegani 2023.)

LLM can perform question-answering (QA) tasks. QA involves retrieving answers from a given text in response to questions and can be used to automate responses to frequently asked questions using a knowledge base or relevant documents as context. There are three main categories of QA tasks: extractive, open generative, and closed generative. Extractive QA involves directly extracting answers from provided context, including text, tables, or HTML. Open generative QA generates free-form text answers based on the given context, allowing responses that go beyond existing information and introduce novel answers. In contrast, closed generative QA generates answers without relying on any provided context, producing responses solely from the model's internal information. (Huggingface a.)

Masked-Language Model

Masker Language Models are powerful AI models that have become popular in natural language processing in recent years. Masked Language Models mask certain percentage words of a sentence and train a model to predict that masked words based on the sentence's other words, as seen in Figure 11. This approach forces the model to learn contextual relations between words. It predicts the precision of the missing words improving the precision and efficiency of general tasks of natural language processing. (Mishra 2021.)

Figure 11. Masked Language Model example (Mishra 2021.)

One of the key advantages of MLM over language models is its bidirectional nature. It considers the context of the masked word on the right and left sides. MLM can capture stile relationships between words and improve the general accuracy of the predictions. Also, MLM can be fine-tuned for specific domains or tasks that can increase even more. (Devlin et al. 2019.)

In addition to its efficacy for specific tasks, MML also be used as a pre-trained general model. This approach simply trains the model with massive data to learn patterns with the subjacent relation between words. These pre-trained models can be fine-tuned for specific tasks with smaller amounts of data to improve performance and reduce train time. (Devlin et al. 2019.)

MLM has been effective for a variety of NLP tasks. GPT-2 and 3 have achieved state-of-the-art results on various text generation tasks. In contrast, BERT and RoBERTa, an optimized BERT approach, has shown to outperform other models such as sentence classification, name entity recognition or question-answer tasks. (Khan 2019.)

Pre-trained model

Pre-trained models are machine learning models built on mathematical algorithms trained using data. Those models have been trained on datasets and fine-tuned with weights and biases. The parameters of pre-trained models are already calculated, so it can be used to use the model or load the parameters and keep training, so the training process is not started from scratch. (Nvidia.)

These models are trained with a dataset using computing resources. Depending on the number of parameters, it can take a long time and computer resources. In addition, AI

training consumes a vast amount of energy. Some causes of this are the requirement of millions or billions of training examples, many training cycles and a high number of weights calculation and mathematical operations. (Numenta 2022.)

This model's usage gives developers an advantage as it is simple to incorporate the model. It gives a solid or better model performance faster than doing it from scratch. As the model's core is already trained, much-labelled data is not required. This model gives versatility for transfer learning, prediction and feature extraction. (Shao 2019.)

Fine-tuning

Fine-tuning is a learning technique by learning transfer that has gained popularity in the last few years. It allows developers to transfer knowledge from a pre-trained model to another model that is specialised in a specific task. Transfer knowledge is accomplished by taking a pre-trained model, loading its parameters, and training it with fewer data but more specific ones. In this way, the model can adapt its characteristics to the new data and improve the performance in the specific case. (Baheti 2023.)

In addition, fine-tuning has several advantages over training from zero. Usually, pre-trained models are trained with data from different fields that gives robust characteristics to generalise when getting a new output and being useful for different tasks. Fine-tuning also allows developers to train a high-performance model with fewer data and computational resources than training from zero. Finally, it allows developers to rapidly adapt preexisting models for new tasks, cutting down time and cost related to the development from scratch. (Sarkar 2018.)

However, there are also some challenges related to fine-tuning. One of the main problems is over-fitting. It happens when a model is trained in a small dataset to make a specific task. Some techniques like regularisation and early stopping are used to mitigate the problem. Pre-trained model characteristics like architecture, task domain, size and complexity should be considered. (AWS a; Sarkar 2018.)

Automatic Speech Recognition

Automatic Speech Recognition is a technology that enables computers to convert audio into text. ASR has become popular in recent years with a wide range of applications. Automatic Speech Recognition uses algorithms to transcribe speech into text automatically. It is also a critical component of natural language processing, allowing humans to converse with computers through voice. (Rella 2022.)

ASR systems consist of three components. The first is an acoustic model that represents how different speakers produce sounds and how they vary depending on different cultural and personal factors. Secondly, the language model component captures the statistical properties of natural language to predict the most likely words or phrases given a context. The last component, the decoder, searches for the best match between the acoustic signal and the language model using different algorithms. (Rella 2022.)

However, as in every field, it is also challenging because human speech is variable and complex. The environment where the audio is recorded has some background noise that interferes with the quality and clarity of speech. Depending on the language, even in the same language, depending on the city, accents can change and different pronunciation of words or sounds changes. The mood, intention or situation also affects speech speed. In addition, the speaker's emphasis can disrupt the sentence's flow and structure. The last challenge is that the words specific to a culture, foreign words, slang, acronyms or proper names that are not part of the system's vocabulary can not be identified correctly. To overcome these challenges, vast amounts of data for different scenarios and domain must be sued for training models. (Rella 2022.)

In ASR research, there are some current trends. End-to-end models aim to simplify ASR systems by combining acoustic and language models into a single neural network to direct map speech signals without intermediate steps. Another trend is self-supervised learning, which uses unlabelled data to train useful representations for ASR tasks without human supervision. The last trend is multimodal models that integrate information from other modalities, such as vision or text, to enhance ASR performance or to enable new applications. (Masuyama et al. 2022.)

ASR has a wide range of applications in various fields. Voice assistants are a clear example where software agents can interact with users through natural language commands using voice input and output. Voice biometrics are systems that can identify individuals based on their voice. Finally, speech-to-text transcription allows the conversion of audio recordings into text documents for several purposes. (Nguyen 2022.)

3.8 Software

AI has become more popular and used in the last few years. As the demand grows, new programming languages and libraries are also necessary. Nowadays, there are hundreds of libraries and languages to develop AI and machine learning tasks, each with advantages and disadvantages. In the following paragraphs, some of the most used ones are described.

- Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. It combines high-level data structures with dynamic typing and binding. Python has a simple syntax that emphasizes readability. It supports modules and packages that encourage modularity. It is the most used programming language for AI development. Some of the good points of using Python is that it has a considerable library ecosystem with a low entry barrier. It is a flexible language, as developers can choose their programming method without recompiling the code. Another good point is that it does not depend on the platform, as it can be executed in different operating systems. (Ryabtsev 2022; Python.)
- R is a programming language that was created for statisticians. As the language contains mathematical computation involved in machine learning derived from sadistic R becomes the right choice. R is a good choice when a project is heavily based on statistics. Similar to Python, it has a bulk of libraries and tools. The language became helpful concerning exploratory work for statistical models. R can be executed in different operating systems. (Kumar 2019; The R Foundation.)
- C++ is a language known for its speed and efficiency. However, it has a complex way of writing code. C++ is less used than Python but is widely used throughout the industry as it is an excellent choice for resource-intensive applications combined with other languages to build applications. (Berkeley Extension a.)
- Julia is another language that can be used for AI. It is designed for high performance as it compiles native code for multiple platforms. The language is dynamically typed and has good support for interactive use. Its ecosystem is designed for data visualization, data science, machine learning, and general purpose. Its packages provide different functionalities, such as the MLJ package for machine learning. (Julia.)
- Jupyter notebook is a web application for creating and sharing computational documents offering a simple, streamlined, document-centric experience. They allow choosing over 40 programming languages, creating a rich interface and introducing massive big data tools. Code is divided by blocs that can be executed independently, allowing developers to run a code sniped each time if needed to test instead of all code. (Jupyter.)
- TensorFlow/Keras are open-source software libraries for machine learning and deep learning models. Tensorflow provides a flexible, high-performance platform for algorithms in different hardware architectures. Keras can be used to design, ex-

periment and prototyping. It acts as a high-level API of the TensorFlow library. (TensorFlow; Keras.)

- PyTorch is a tensor library for deep learning optimized for GPUs and CPUs. One of its main features is stability. It also keeps beta features well-tagged, so anyone can decide whether they want to use them. The last feature is prototyping. It is a similar library to TensorFlow/Keras developed by Meta. (Pytorch.)
- Transformers is a library that provides thousands of pre-trained models to perform tasks on different modalities such as text, images and audio. Transformers provide an API to download those models and a way to fine-tune them with specific task datasets. Transformers is backed by Pytorch, TensorFlow and Jax, allowing a straightforward way to train and load them. (Huggingface b.)

3.9 State of art

AI is a rapidly evolving field that has significantly improved over the past years. It has hugely impacted industries and will change them more in the following years. Following this, we are talking about some newest AI applications.

One of the most notable advances in AI was GPT3, a language processing model that can do impressive tasks such as writing short histories coherently with a topic they have indicated. It can be used for chatbots to keep a conversation, achieved with GPT3.5, released at the end of 2022. OpenAI launched ChatGPT, an AI that can keep a conversation. (OpenAI 2022.)

Another significant milestone in AI is image generation. DALLE-2 and Midjourney are new AI systems designed to assist humans in the design of things. Writing down some ideas and what kind of art they want to generate, the AI will give an image. There are some incredible examples as the one shown in Figure 12. The image is named *Théâtre D'opéra Spatial* and won an art prize (Roose 2022).

Figure 12. AI generated picture that won an art prize (Allen 2022)

Despite these fantastic achievements, AI is still growing fast, and there is a long road to achieving general AI. We will likely see significant progress in this area in the coming years, as could be GPT 4 or a new version of DALLE-2. (Marr 2023.)

4 Front end

4.1 Definition

The front end is everything that a user can see when using an application. It includes visuals, interactive elements, layout, and theme. It is also known as the client side of the application, as it is the part with which the user can interact. Developing a front end includes the design, prototype and programming faces. (Christensoon 2020.)

The main goal of front-end development is to create a smooth and intuitive user experience interface. It has to be adapted to different platforms as it can be used on a computer, mobile phone or any other device with a screen. The layout also has an important role when we want to achieve an intuitive user experience, as most websites use similar layouts that allow users to use them efficiently, as it is shown in Figure 13. (Babich 2019a; Indeed 2022.)

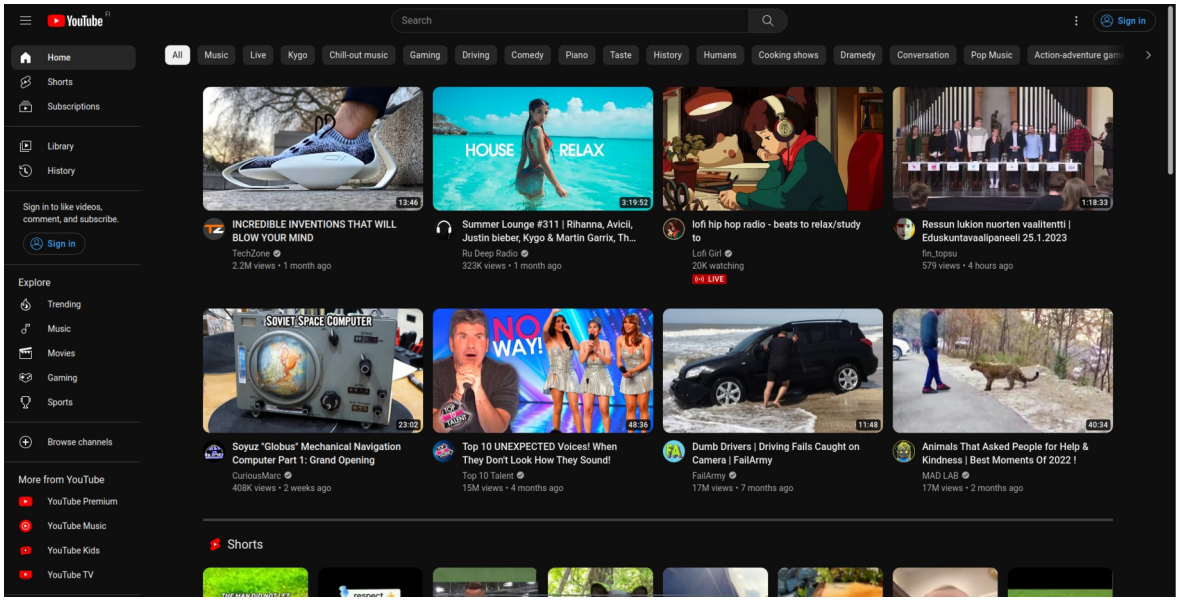

Figure 13. Example of cards layout that most websites use on content-heavy sites

So, the front end plays a crucial role in the success of an application, as it is the primary way the user has to use it. In designing an application, we aim to create a positive experience for the user. (Berkeley Extension b.)

4.2 Types of applications

An application is a computer program designed to perform a specific function. It can be classified regarding the interface. Applications can be classified depending on where they are executing and their appearance. (GCF Global.)

A web application is the most common nowadays. It is accessed through a web browser and can be used in any operating system that has a browser. The application will be accessible anytime, including on mobile phones. Another good point is that the application does not have to be installed on the device to be used, as it is accessed with the browser. This characteristic leads to a more cost-effective option when discussing the development process. (Martin 2023.)

A native application is built for a specific operating system. One of its characteristics is that its design looks and feels like a part of the system. Also, they are programmed to communicate directly with the operating system resulting in faster apps. However, they can be more costly and time-consuming to create and maintain as a version for each operating system is needed. (Ijaz 2022.)

A hybrid application contains elements from native and elements from web apps. It is deployed using a web view container to display web content inside a native application. As a result, it can execute native code to access specific hardware only native applications can access. This kind of application allows developers to combine the user experience of different platforms in a single application with controlled costs, as code can be reused for different platforms. It reduces the development time and costs as just one version of the application must be developed and maintained. However, only some things are good. The interface does not feel native and can not use full platform capabilities producing a slow performance and a dependence on a browser. (StarDust Testing.)

The last type of application is cross-platform, designed to run on multiple platforms. They use frameworks to achieve the native app experience. As a hybrid, this allows low-cost development with a single code for multiple platforms. However, there is a limitation on consistency with native UI components and a high on performance and hardware compatibility problems. (Manchanda 2022; Kotlin 2023.)

In conclusion, the application can be classified by its interface. It is necessary to determine the requirements and objectives to choose the right type of application for the project. Figure 14 provides a visual representation of this classification.

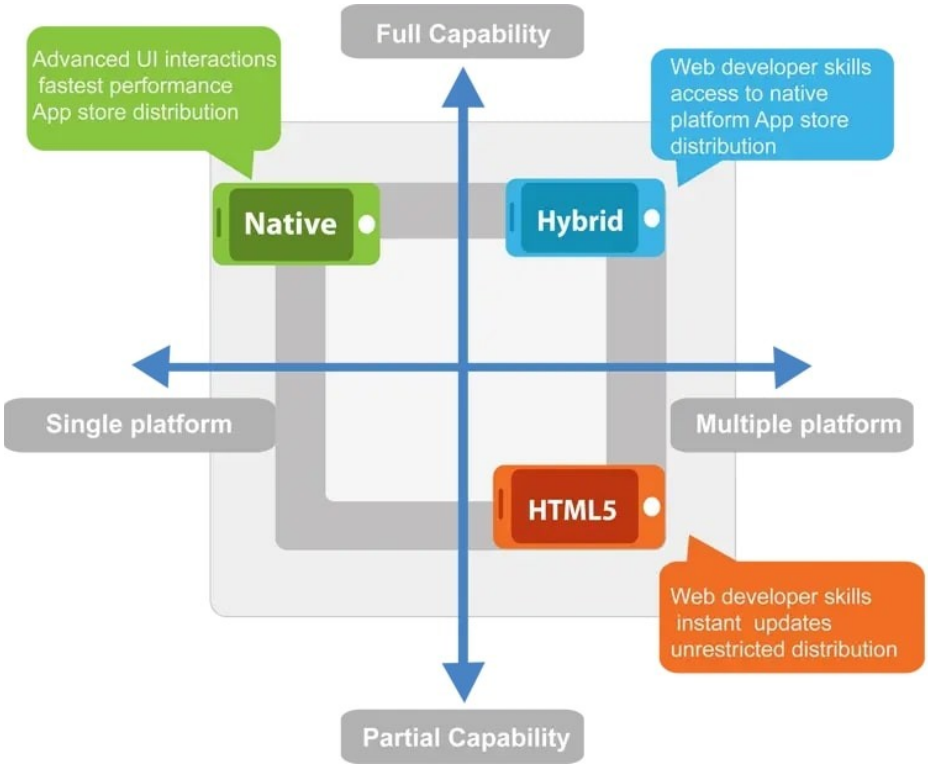

Figure 14. Native vs Hybrid vs Cross Platform Applications (Mahajan 2020.)

4.3 User Interface

Human-Computer Interaction is a multidisciplinary field focused on the interaction between humans and computers. It incorporates elements from different disciplines, including computer science, cognitive science and human factors engineering. These different fields work together in HCI to provide a way for humans to interact with machines. The main goal of it is to understand and optimize these interactions. (Interaction Design a.)

HCI includes different elements that intersect between them, human and computers. This field allows users and computers to communicate, mainly focused on interfaces. As nowadays there are computers in every industry, it is important to consider the design of the interfaces. In consequence, HCI includes user-centerd design, user interface design and user experience. This areas have a directly impact over the interface usability of the interfaces. So this areas are essential to be considered as factors when a interface is being design. (Fawcett 2021.)

User centred design

User-centred design is a primary principle of HCI that uses the user as the centre of the design of an interface. As they will be using it daily, it is essential that users feel comfortable and confident using it. (Babich 2019b.)

During the design process, it is essential to understand the client needs and goals of the users and the context of where the interface will be used. It can be accomplished through their active participation in the design and testing process. It also includes making users perform specific tasks using the interface. Some parameters will be measured and considered in the interface's design process. (Toczyska 2018.)

When developing an interface, developers have to know what the client has requested the application to do. So, functional requirements, what the system must do, and non-functional limitations of the system and its development must be considered as a clear list. They will only implement necessary things the client wants or pays to be developed. (ReQtest 2012.)

Design guidelines

Design guidelines are suggestions for using design principles to provide a pleasant user experience. This advice incorporates, among other things, intuitiveness, efficiency, consistency, mistake avoidance, and feedback. They also address design features like components, layout, style, accessibility, and text. (Interaction Design b.)

Companies are expected to apply these criteria since their products are often identifiable by them. Guidelines offer principles that assist users in avoiding annoyance when using the interface. In addition, impairments and the environment of usage are considered. The following are some particular guidelines worth considering: (Kimbarovsky 2023.)

- Simplicity: Remove unnecessary elements that do not have a functional purpose. Avoid using too many colours, and use a highly legible font. The main point is to avoid being over-saturated with graphic elements. (Juviler 2021.)
- Visual Hierarchy: Arrange and organize elements in a way that helps users naturally gravitate to the most critical elements. An interface has to be designed to make it easy for users to complete their actions naturally. (Juviler 2021.)
- Navigability: Keeps primary navigation simple, use breadcrumbs to remember the navigation trail and uses a search bar if many elements can be visited in the application. A footer navigation can also be added. (Juviler 2021.)

- Consistency: This keeps an overall look and feels across the entire app, but the layout can be changed on a specific page. It makes the user understand where is the information they are looking for. (Juviler 2021.)
- Responsivity: The ability of an app to adapt to different sizes of screens in a way that maintains a great user experience without overlapping elements. In some instances, it will automatically resize and shuffle to fit the dimension of the device where the application is being used. (Juviler 2021.)
- Accessibility: Make the app for everyone. It includes people with disabilities or limitations that affect the browsing experience. A designer have to think about the structure, page format and visuals. The interface should be robust to avoid critical errors and be operable in different situations. (Juviler 2021.)

Testing user interface

Before releasing a product, we have to test and evaluate our work. It is also the case for user interfaces. In this case, testing takes site during the design process to ensure that the interface is effective, efficient and user-friendly. There are different techniques to evaluate interfaces, including: (Hamilton 2023.)

- Usability testing: A user tests the requirements of a website. It also includes tests where the user has to do specific tasks, and time and errors will be measured to improve the user's satisfaction. (Hamilton 2023.)
- Record and Replay: Includes automation tools that record steps to do certain tasks. Records are executed again to check if a task can be done. (Hamilton 2023.)
- User interviews and focus groups: Involves gathering feedback from users through discussions. This process can help identify necessities, issues or areas of the interface that need improvement. (Hamilton 2023.)

By testing the interface, designers can identify and address problems, improving the final product. This process results in a better interface experience on the usage of it by the user. (Hamilton 2023.)

4.4 Mock-up and prototype

*A mock-up is a model of something, which shows how it will look or operate when it is built, or which is used when the real thing is not yet available (*Cambridge Dictionary). In the context of graphical user interfaces, a mock-up is a static design of an interface that

replicates the final design without being functional. It provides the client with a visual draft of the various pieces in the interface in the specified layout. It is used at the beginning of development to test ideas. It allows us to make adjustments without incurring high costs and keeps stakeholders informed about how the final product will feel. A mock-up also allows the client and stakeholders to give feedback about it. (Hufford 2022.)

Another tool is prototypes. It is early adoption of the product that includes some degree of functionality. One of their main advantages is that code is not needed in the design. Some tools, such as Figma or Penpot, allow designers to create moke-up or prototypes. (Sketch 2022.)

Prototypes' advantages are that they allow designers to test functionality, navigation and other aspects of the interface. It allows us to know how the interface will work with high fidelity. However, one of the main problems with this is that stakeholders usually think it needs to be done, leading to misunderstandings and unrealistic expectations. The differences between these three designs can be seen in Figure 15. (Sketch 2022.)

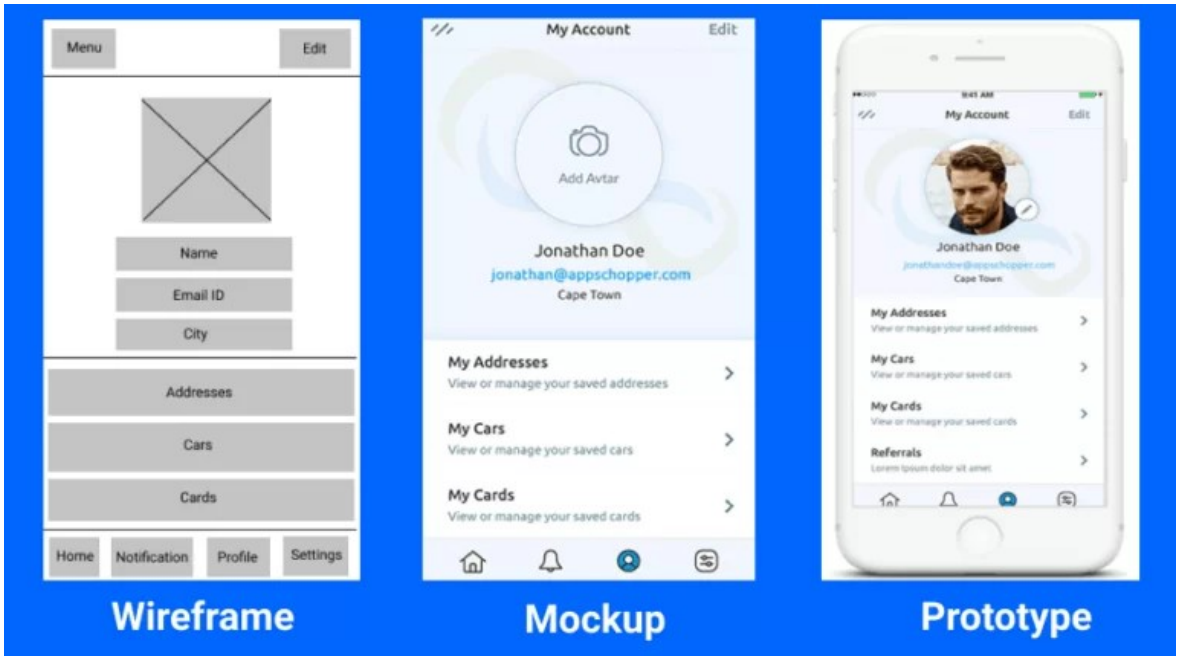

Figure 15. Wireframe vs Mockup vs Prototype (Appshopper 2021.)

4.5 Languages and frameworks

HTML, CSS and JavaScript are core technologies for building web applications. HTML is a markup language used to structure and organise content. Markup *is a set of detailed instructions, usually written on a manuscript to be typeset, concerning style of type, the makeup of pages, and the like* (Dictionary). HTML can be considered a language to create instructions about style, format, structure and type. HTML helps developers to structure pages into elements such as buttons, paragraphs, and navigation bars. Another language is CSS. CSS is considered a design language that stands for Cascading Style Sheets. It is used to make styles that make websites more attractive to end users. The last language of this set is JavaScript. It is a programming language that allows the implementation of complex features on web pages. It allows the creation of dynamically updated content and control multimedia, among others. (Ubah 2021; Mozilla 2023.)

Angular is a development platform built on Typescript. It includes a component-based framework for building scalable applications. It has a collection of libraries that can be integrated easily, including features such as routing, forms management, and client-server communication as the most important. (Angular 2022.)

Flutter is *an open-source framework by Google for building beautiful, natively compiled, multi-platform applications from a single codebase* (Flutter). It uses Dart as a programming language that makes development faster and easier. Flutter includes hot reload for developers that make changes appear fast in the application without restarting it. The application is compiled into native code, giving the best performance possible. It also includes elements with the same principles as Google's Material Design guidelines that are expressive and flexible. (Adservio 2022.)

Vue is a JavaScript framework for building user interfaces. It uses HTML, CSS and JavaScript. One of its main characteristics is the component-based programming model and declarative that helps developers to develop user interfaces efficiently. Some of the two core features of Vue is declarative rendering. It extends standard HTML with a template syntax that describes HTML output on JavaScript state declaratively. It also allows reactivity that tracks JavaScript state changes and updates DOM when changes happen. Vue is designed to be flexible and adaptable. Some cases where Vue can be used are single-page application, server-side rendering, desktop, mobile, WebGL or event terminal applications. (Vue.)

React is a JavaScript framework and library used to build user interfaces and interactive web applications in a fast and efficient way created by Meta. Developers create encapsu-

lated components that manage their state. React focuses on the user view and divides complex interfaces into simple components, increasing the re-usability and execution of the visual representation. React combines the fast and efficiency of JavaScript with a practical methodology to manipulate document object model that allows for building dynamic web applications. In addition, React is declarative and component-based, allowing a predictable and straightforward model for debugging. It also allows the development of mobile applications with React Native. (React; Herbert 2022.)

5 Back-end

5.1 Server

Servers are defined as high-power computers that process, store and manage devices. It works by requests over the network in the structure of the client-server. These kinds of computers are essential for a business to hold its services. We can find servers everywhere. Every website has a server behind it. This way, someone can access it anytime. Another example is the email that every time a user sends, one calls a server to do the process. That is why servers are built efficiently to handle a large volume of requests from multiple clients. (Ingalls 2021; Computer Hope 2022.)

Comparing desktops with servers, find key differences that characterise the latter. One of the primary advantages is scalability, that allows to increase the number of users, devices, or computational power easily achieved by specific hardware. Some of them are high processing power CPUs and high amounts of ram and other hardware components such as TPUs or GPUs. Usually, servers are critical structures as they hold services, so reliability is an important characteristic. It is accomplished with several servers that offer the same service, so we keep a level of failure tolerance. The last important characteristic is collaboration. It allows multiple people to access the data at the same time to the server to work with or share data at the same time. (Ingalls 2021.)

Type of server

Several services, known as software servers, can be used for different purposes. These servers provide services to clients over a network. In the following paragraph there are some examples:

Web servers are used to access websites. They store the information on the website and send it to a client's computer when accessed. Other ones are virtualisation servers are specialised in hosting virtual machines. They allow the connection and working in a personal system. Also, on the same server, people have different operating systems customised and with specific applications. Server operators use a hypervisor to run multiple virtual machines on a single server. Application servers are specific ones for hosting applications. They allow clients to access them using the internet. They hold every specific data necessary for the communication between client and server, making it more manageable. The last type of server is a specialised database that stores a vast amount of data that an application needs or is creating. This kind of server can hold more than one instance simultaneously. (Indeed Editorial Team 2023.)

Operating System

An operating system is a system of software that initialises when the computer is opened and manages hardware and software-like services. Several types of operating systems can be used. In this case, we focus on the server ones. Each of them has its features and characteristics. The following operating systems can be found:

Windows server *is a group of operating systems designed by Microsoft that supports enterprise-level management, data storage, applications, and communications. Previous versions of Windows Server have focused on stability, security, networking, and various improvements to the file system. Other improvements also have included improvements to deployment technologies, as well as increased hardware support*. (Microsoft 2022.)

Linux servers are open-source, so they are free. They are designed for multi-user, multi-process and multi-thread environments. However, this kind of server requires more technical knowledge. As they are open-source, everyone can review their code. It is easy to find and fix security flaws. We can use a terminal or a graphical interface to manage this system. The most important ones are the Ubuntu server, Debian server, Fedora server and Arch Linux. (Marijan 2022.)

The last type of server is Red Hat Enterprise Linux (RHEL). It is a paid Linux distribution with extensive support that provides a fast solution to security vulnerabilities. The good thing is that some RHEL-compatible Linux systems are free, such as Rocky Linux (successor of CentOS) and Alma-Linux. (Marijan 2022.)

5.2 API

An application programming interface or API is a set of definitions and protocols that enable software components to communicate between different services regardless of implementation. Communication is done between the client, making a call and the server that gives a response, as can be seen in Figure 16. API can be thought of as a mediator between users and the resources that they want to get. A REST API is an application programming interface that adheres to the design principles of the representational state transfer architecture style, as Roy Fielding defined it for the first time. The REST API offers developers considerable flexibility and freedom. (IBM f; Red Hat 2020.)

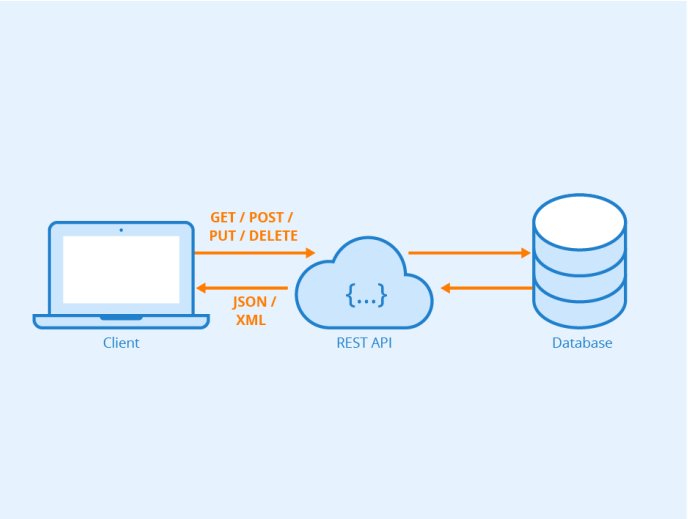

Figure 16. REST API (Seobility.)

RESTful APIs offer several benefits, including scalability, flexibility and the independence of the technology used. The RESTful design allows scalability with the optimization of the client-server interactions. Removing the charge of the server with the absence of a state and using the cache with proper management can be used to remove some client-server interactions. In addition, RESTful web services accept the total separation of client-server. Separation allows components to evolve independently. The platform or technology changes on the server application do not affect the client application, and the capacity to add layers to the application's function increases the flexibility even more. REST APIs also are independent of the technology used, allowing developers to write client applications and servers in various programming languages without affecting API design. Additionally, the change of underlying technology on either side does not affect communication by HTTP, URLs and JSON or XML. (AWS b.)

REST APIs create, read, update and delete operations over a server. It is crucial to ensure to follow security standards to avoid vulnerabilities. There are different ways to improve security. Implement authentication that validates the client that is calling the API. OAuth 2.0 is a standard that allows users to access resources from third-party applications. Tokens do the access. Transport layer Security encrypts the information so it can not be read. Also, parameters passed from client to server must be validated by establishing a schema and validating parameters against it. An API also handle a large number of requests within a short time. The number of calls a client can make in a specific time can be limited. It is called rate limiting and is used to stop brute-force attacks. Retrieving in-

formation from an API can return large amounts of data. To limit the resources that can be retrieved in a single call are limited by pagination. (Bilal 2022.)

Developers need documentation in order to explain how the tool works. With API documentation, developers can understand what it is possible to do with the API and how to use it. A straightforward way to follow documentation allows an easy adaptation and a better experience for the developer. They are allowed to plan and build the integration without external support efficiently. Some characteristic of documentation is the interactive calls where developers can test how it works before implementing it on their product. (Garcia.)

The process of testing an API verifies that it accomplishes all established requirements. Testing can be done manually or automatically and is considered part of the integration test. Manual testing involves direct interaction with an API using different tools, while automated testing uses specific software to send calls to the API to compare results with the expected result. Testing an API helps to ensure that different components of a system work as it is expected. Several principles must be followed to ensure the quality of an API implementation, such as API tests in the continued integration and small specific tests. (Nyakundi 2022.)

5.3 Message Brokers and Task Queue

A message broker is a software that allows communication between applications in a distributed way. Brokers are intermediaries between different applications and services to send data in real time between applications even if they are not online. (IBM g.)

Brokers use a communication pattern based on publish and subscribe events. There are different components of a message broker. Produces is responsible for sending the messages to or publisher. The consumer is the one that consumes messages in the message broker. It is the subscriber. The last component is where the message broker stores the messages queue. (Dhanushka 2020.)

This system ensures the message is delivered even if the consumer is inactive. When the consumers return, it can start consuming the stack of messages stored in the queue of the message broker. As the message broker is asynchronous, processing tasks are done separately and do not affect the main thread. Applications can be programmed in different programming languages but can communicate easily. (Subhashana 2021.)

A task queue implements a message broker system that allows the execution of asynchronous tasks. They are used to distribute work across threads o different machines. The

queue is composed of units of work called tasks. Some workers monitor the queue waiting for new work to perform. Some task queue uses message brokers to communicate via messages between client and workers. It allows the tasks to be accumulated in a queue so the worker can consume them, or even if the worker is offline when it comes back can consume them. It can be accomplished by exposing an HTTP endpoint and having a task that requests it. (Celery.)

5.4 Languages

Tools and programming languages are essential for developing software applications. Developing APIs and choosing the right tools, such as message brokers and task queues, can make a big difference in the functionality and efficiency of the resulting software. In the following paragraphs, some tools and languages are explained.

Python and JavaScript are can be also used to develop an API, as explained in the previous points. PHP is a popular server-side scripting language for backend development. PHP code is executed on the server, and the result is returned to the web client. PHP is designed to work perfectly with HTML. It is also known for its capacity to connect to databases making it a powerful choice for creating web applications powered by databases. It has built-in features to create RESTful APIs, and PHP is a popular choice for its ease to use, flexibility and large community development. (PHP.)

Rust is a programming language that runs fast, prevents segmentation faults and is memory safe. It is designed to build an efficient app and is growing in the web development field. It is designed to be a low-level language to offer performance and control with modern programming concepts such as safe memory management, concurrent and zero-cost abstractions. (Rust.)

FastAPI is a modern, high-performance web framework for building APIs with Python 3.7+ based on standard Python-type hints. It is designed to be used to use and have a high performance, allowing developers to create APIs in a fast way. FastAPI includes automatic validation of data and automatic document generation and allows asynchronous code. It is also compatible with a vast number of databases or web servers. (FastAPI.)

Actix is a robust, pragmatic and high-speed web framework for Rust. It provides a powerful and flexible to build distributed applications. It is designed to be easy to use and flexible, with abstractions for handling HTTP requests and responses that allow building RESTful APIs. It also works as a middleware for handling everyday tasks such as authentication, logging and compression. (Actix.)

NestJs is a progressive Node.js framework for building efficient, reliable, and scalable server-side applications built on the typescript. It is designed to provide a set of abstractions to make it easy to build applications. NestJs supports microservices and WebSockets. It focuses on modularity and extensibility with built-in support for various tools and libraries. (NestJs.)

Celery is a task queue library written in python that enables distributed task processing. It can be used with other languages through message broker implementations. Celery allows the execution of tasks asynchronously in the background, so it is usually used for time-consuming tasks. It also provides features such as task scheduling, result storage and monitoring. Flower is a user-friendly monitoring tool in real-time. It provides a dashboard that shows information about workers, queues and tasks. Flower also allows controlling the task execution. (Celery; Flower.)

RabbitMQ is a message broker software that implements several transport message protocols. It is a lightweight and easy-to-deploy system developed on Erlang. RabbitMQ is designed for clustering, failover, high availability and scalable. (RabbitMQ.)

Redis is an in-memory data structure store that can be used as a database, cache and message broker, supporting many data structures. Redis has a high performance and is scalable, with the ability to cluster and replicate for high-availability situations. It is written in C. (Redis.)

6 Database

6.1 Definition of database

A database is a collection of structured information stored and accessed electronically. It allows the manipulation of data quickly and efficiently. The use of a database management system dates back to 1960. (Oracle.)

Most people think that a spreadsheet and a database are the same. However, there are some differences. A single user uses a spreadsheet to store and manipulate a small amount of data. A database is designed to be used by multiple users, holding a considerable amount of data with logic operations being executed on it with a specific programming language. (GeeksForGeeks 2020.)

A DBMS is a software that stores, retrieves, and runs data queries. It is used as an interface between the user and the database, allowing data modification. It also allows the manipulation, extraction and visualization by users and other programs. (Wickramasinghe & Raza 2021.)

6.2 Types

When talking about databases as engineering, we can decide between two types. The main difference between them is how they are structured and how they store data.

On the one hand, SQL or relational databases are designed to store information in a table. In each table, we can find rows representing an object and columns indicating the name of the object's property. The values found in a column must be of the same type. SQL language can be used to manage data, such as getting or adding a new one. SQL databases follow ACID principles, atomicity, consistency, isolation and durability, ensuring data are kept consistent. (GeekForGeek 2022.)

On the other hand, NoSQL databases, or distributed databases, offer a more flexible way to store and retrieve data. Information is stored in a flexible way that the designer defines. Each database has its unique structure. If we need to increase computational power, we can add new servers thanks to the flexible schema and horizontal scaling provided by NoSQL databases. It is easier for developers and allows fast queries due to the data model. NoSQL follows CAP principles, consistency, availability and partition. This database allows fast-paced, agile development and storing of a large amount of data. The differences can be seen in Figure 17 (MongoDB.)

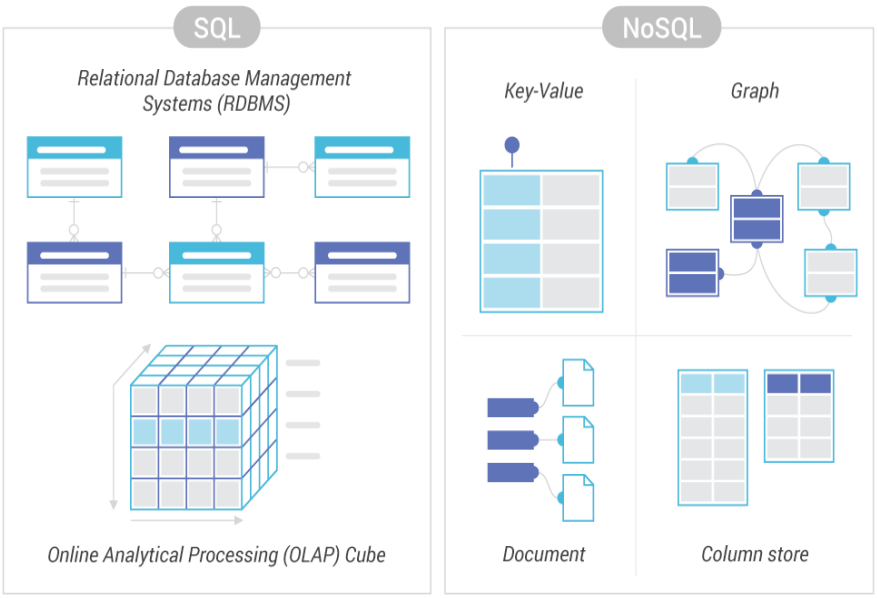

Figure 17. Differences between SQL Database and NoSQL Database (Scylladb).

6.3 Design

Database design is an essential part of developing any application involving data. It includes data management systems design, development, implementation and maintenance. (Peterson 2023.)

The first step in database design is requirement analysis, which involves planning the requirements and the definition of the system to limit the scope of the development. After defining the requirements, the engineer has to start thinking about the design of the database. It involves creating the structural model that establishes the structure of the data elements and the relationship between them. The logical model is based on entities and the relationships between them, entities' attributes and characteristics. It is a graphical representation of the requirements of the product. (Naeem 2022; Tylor 2023.)

After the logical mode, a physical mode has to be created. It describes the implementation of the logical model. The physical one includes details such as columns, keys or data types, and it is designed for a specific DBMS, including all the details of the relations and user profiles. (Naeem 2022.)

Finally, the last step is implementation. If the company migrates from an old system, old data must be converted to the new scheme and stored in the new database. Also, the database has to be tested to identify if there are any errors. (Naeem 2022.)

7 Practical case: Learning Educational Application

7.1 Introduction

Now that most related concepts are explained, especially back-end and natural language processing, it is time to put concepts into practice. In developing this project, the main objective is to develop and test a prototype of how artificial intelligence can help students with their education. In this system, the objective is to create a REST API that allows the creation of the necessary resources to transcribe videos and answer questions. The system allows the upload of teaching resource videos that are transcribed. With it, students can formulate questions that the system answers. The functionality is accomplished using artificial intelligence. The face problem relates to natural language processing, precisely automatic speech recognition and question answering.

In this case, the use of artificial intelligence suits the case of question-answering tasks. There is no way to do it without predefined questions in the system. With this method, the answer is given by the text provided by the different resources uploaded. It is hard for this type of artificial intelligence to respond well, as the good ones are behind huge companies such as OpenAI, Google or Meta. The transcription task could be done without the help of artificial intelligence. However, this implies lower accuracy and efficiency and explains the usage of artificial intelligence.

Firstly, a REST API must be developed that includes a way to create subjects and resources. An API allows the creation of a programming interface that developers can use to create applications, as explained in the theoretical part. Several endpoints are created, and good documentation is made in order to help developers to implement it in a fast way without the need for external support. As everything is computed locally, tasks generated by the API are stored in a task queue, allowing the system to keep up and not use all resources, leading to a crash. This queue will handle transcription and question-answering tasks as they need to load heavy models into the video card memory. In order to test the system quickly, a web application contains three pages, two to create and list subjects and resources and a last to see the watch information.

7.2 Project planning

The planning and organization of a project are crucial to achieving a final product. A well-organized strategy ensures that all aspects of the project have the needed attention and that the product meets the desired goals and objectives. In this project, iterative and incre-

mental development method is used. This approach breaks the project into smaller parts, allowing for flexibility and adaptation throughout development. Developing divided by versions is an effective strategy for organizing the project's progress. It gives the ability to track progress in a better way.

In addition, breaking the project into stages is essential to create a clear and detailed plan for development. This plan includes milestones and resources required for each stage of the project. It also allows provisioning versions of the project so the client can see how the development is going and use the early stages of the project.

After deciding what method is used for the development, it is time to consider the project's different parts. As the project aims to use new artificial intelligence models, they must be searched for and tested to find which fits the project's necessities better. In order to test the viability of the idea, a small prototype of only artificial intelligence is developed.

The following step is to think about and develop the backend. It is considered the engine that powers the system. It is responsible for processing data and executing tasks. It is essential to design the backend well and choose the right resources and endpoints for the project needs. After resources and endpoints have been chosen, it is crucial to write them down and document the functionality in each endpoint to keep the project concise so unnecessary things are not implemented. It is also important to document the backend well so developers can quickly test it.

Finally, the last step is the development of the frontend, which is what the user sees and interacts with when using the system. Considering the target audience's needs and the system's finality is important. In this case, the frontend is a graphical interface in order to use the RestAPI designed before and is not supposed to be a final product to be released in production or institutions. When starting the development process, a prototype is made to see where different functional parts of the interface are located. After defining the final interface, the coding part starts.

Effective planning of a project is essential to ensure its success. The selection and integration of the appropriate AI model, the backend's design and documentation, and the frontend interface's development are critical steps that require close attention. The specific points also explain the election related to the tools used. Each detail is explained in the following points and the decisions that were taken explained that led to the result of the system's development.

7.3 Artificial Intelligence

As explained, the project's core is using artificial intelligence. Given the importance of adequately selecting what model should be taken, some tests are done. As the project needs two artificial intelligence, the first one that has to be chosen is an automatic speech recognition model that transcribes audio to text. The second one is the one in charge of the question-answer task.

Related to the first task, suppose we want to use the best method possible. In that case, we have to consider every possibility that exists on the market and can be executed in the available hardware. Starting with traditional methods, hidden Markov models and dynamic time-warping speech recognition mainly use statistics and probability theory to recognise and transcribe language. However, as explained above, these traditional methods have several drawbacks when different accents, speaking speeds, words and noise are found in the audio. The noise can be fixed partially with a digitalised spectral gating algorithm similar to a noise gate.

In the last year, artificial intelligence development has stepped up. Some state-of-the-art models have appeared, such as OpenAI Whisper, Talon model and Nvidia NeMo Model. These models incorporate deep learning techniques based on transformer architecture or neural machine translation techniques. As the project's objective is to use AI, one of those three is used. In the first case, we discard Talon as it is a close source model that is not accessible. In the second case, NeMo is an open-source toolkit that makes many things related to voice and conversations, which is excellent. However, the preference of taking a model that makes a thing well prevails. Whisper is an open-source model for automatic speech recognition.

Despite this choice, different metrics should be compared. The models often have different sizes. In this case, Whisper has eight different versions, Talon and NeMo two. Three parameters are taken to make the comparisons correctly, and only the large model. The results are shown in Table 2.

Table 2. ASR model comparation (Google Docs 2022)

Dataset	Whisper large-v2			Talon d-1B			NeMo xlarge		
Metrics	WER	Perfect	Useless	WER	Perfect	Useless	WER	Perfect	Useless
common voice v10	9.63	60.38	2.06	9.44	59.24	1.13	5.78	72.34	0.91
librispeech clean	2.72	72.69	0.23	2.45	71.30	0.04	1.49	81.47	0.04
librispeech other	5.26	57.55	0.68	5.46	53.49	0.34	2.82	70.19	0.03
mls	7.85	24.22	0.43	8.04	16.53	0.00	5.14	27.25	0.00
accent2-all	6.12	20.78	0.34	10.45	7.57	0.23	8.54	13.02	0.72
tts1	25.34	47.12	27.16	4.81	84.13	4.30	33.82	44.08	15.83
tts2	54.32	60.23	39.17	24.75	79.04	20.28	74.63	45.65	53.22
words1	16.70	83.71	14.06	7.55	91.54	6.97	31.85	71.46	25.24
words2	88.11	52.90	47.08	22.26	77.76	22.24	136.06	37.00	63.00
words3	26.23	72.51	20.44	16.16	79.89	13.36	28.92	68.68	23.21

WER: Word Error Rate for each model per dataset (Lower better)

Perfect: Percentage of transcripts with no errors for a dataset (Higher better)

Usless: Percentage of transcripts with >75% word errors for a dataset (Lower better)

After analysing the results, Talon and NeMo have some outstanding metrics but other bad ones depending on the dataset. However, Whisper model is the one that performs on average well in every dataset. In addition, this model can also detect the language the speaker is talking about and translate it into another language. So for this project, Whisper AI is the model used for ASR.

In order to test it, Google Colab notebooks are used. Figure 18 is a code block designed to transcribe speech from an audio input. It loads a pre-trained neural network model from Whisper AI and sets up various audio decoding options. The beam_size parameter controls the number of candidate transcriptions generated by the model during decoding and the temperature that introduces the randomness of the decoding process.

After loading the model, and decoding options set the code to transcribe the audio using the task parameter and transcribe functions. The model produces an output as a string that can be codified as JSON. This output provides the text, language and an array of

segments that gives the text start and end, among other parameters that are not that important. The video or audio can be uploaded into Google Colab and loaded by the path. There is also the possibility of downloading a YouTube video with the YouTube-DLP library. This code is only for testing purposes and is a way to test the different size models that Whisper provides.

```
model_name = "large"
model = whisper.load_model(model_name, device="cuda")

beam_size=5
best_of=None
temperature=(0.0, 0.2, 0.4, 0.6, 0.8, 1.0)

decode_options = dict(language=lang_picker.value,
                      without_timestamps=False,
                      best_of=best_of,
                      beam_size=beam_size,
                      temperature=temperature)
transcribe_options = dict(task="transcribe", **decode_options)

transcription = model.transcribe(audio_picker.value, **transcribe_options)
```

Figure 18. Whisper AI load and transcribe test

The second AI task that the project has to solve is question-answer. The main idea behind this is that the AI answers the question by extracting the answer from the context, leading to a limitation of the generation of free text or answering things that are not provided in the context. Depending on the project's aim, the model can be restricted to a specific domain leading to more precise answers or an open domain involving several ones.

Defining how the task needs to be solved is crucial in identifying the appropriate model for addressing it. As other ways to solve the problem involves huge models that can not be executed on a local machine, the model type used for the task is extractive question-answer. The characteristic of this variant is that it extracts the answer from the context, so questions not found in the text can not be answered.

Regarding question-answering models, one that stands out is BERT (Bidirectional Encoder Representations from Transformers). Developed by Google, BERT has been fine-tuned for various NLP tasks, including sentiment analysis, question-answering and text generation. Its success is mainly due to its ability to capture bidirectional context information and use a masked language modelling approach during pre-training.

Another powerful language model is Roberta, which Facebook AI Research developed as a variation of BERT. Like BERT, Roberta is based on transformer architecture and has achieved impressive results on various NLP benchmarks. However, Roberta uses a more extensive training corpus and a different pre-training objective than BERT, which has led to improved performance in a range of downstream tasks.

After exploring different AI models, Roberta-base-squad-2 was determined to be the best choice for our project's question-answering task. It demonstrated superior performance in extractive question-answering, which involves directly extracting the answer from the given context. Furthermore, Roberta-base-squad-2 showed exceptional accuracy in answering questions in a specific domain, making it a perfect match for our project requirements. In addition, the model's relatively small size meant that it could be run on a local machine without any performance issues, which made it an efficient and cost-effective solution.

Before implementing it on the backend, the model has to be tested. As in the Whisper model, the code is done in a Google Colab notebook. In Figure 19, the code imports the transformer library used to initialize a pipeline. It creates a pipeline for question-answering tasks introducing the model name and tokenizer. The model used is defined by the name deepset/roberta-base-squad2. Deepset is the name of the company that has trained the model, and roberta-base-squad2 is the name of the model that combines the base one used and the dataset for fine-tuning. After setting up the pipeline, a question and context are introduced and provided to the pipeline that returns the answer to that question with some parameters that give an idea about how good the answer is.

```
from transformers import pipeline

model_name = "deepset/roberta-base-squad2" #Name of the model

print(model_name)
nlp = pipeline("question-answering",
               model=model_name,
               tokenizer=model_name) # Setup pipeline
#@title Question and answer
question = "What is blender?" #Question here
QA_input = {
    'question': question,
    'context': transcription['text']
}
answer = nlp(QA_input) # Run inference
print(answer['answer'])
```

Figure 19. Question-answer test

After introducing both tasks of transcribing and question-answering, the combination of both leads to a small playtest notebook that can be used to test and experiment with the system on a small scale. It can be achieved using a Google Colab notebook, which allows for easy execution of both artificial intelligence. Using this notebook, one can easily download a video file from Youtube, transcribe it with Whisper, and then use the transcribed text as input for the question-answering pipeline. This simple setup allows for testing and experimentation before implementing the system into a larger backend, making it an efficient and effective way to refine the system and improve its performance. Additionally, Google Colab provides access to powerful GPUs, enabling users to experiment with larger models and datasets for improved accuracy and performance.

Related to the programming language, Python is the one used as it is an ideal language for implementing artificial intelligence solutions. It has a massive community behind it with numerous libraries and tools that make work easier, like transformers, yt-dlp and Whisper libraries.

7.4 API development

Once the models are decided, the technologies used for the backend have to be decided. It is crucial to consider the technologies used as they directly impact the API's performance, scalability and maintainability. Evaluating different technologies and deciding based on factors such as the project's specific requirements is crucial.

Before starting the development of the API, it is essential to have a well-defined design. The API's core functionality and structure are defined in the design process. Endpoints and resources must also be defined during this process, as seen in Figure 20. By designing the system, problems can be faced before implementing, making changes less costly. In the system, three primary endpoints hold functions related to each resource. Subject, resource and media are the ones that the API needs to manage the needed information of the resources. These functions change the information over the resources in the database. After doing the design, it should consider that it is a design and can vary during the development process because some necessities may not be fulfilled with this design.

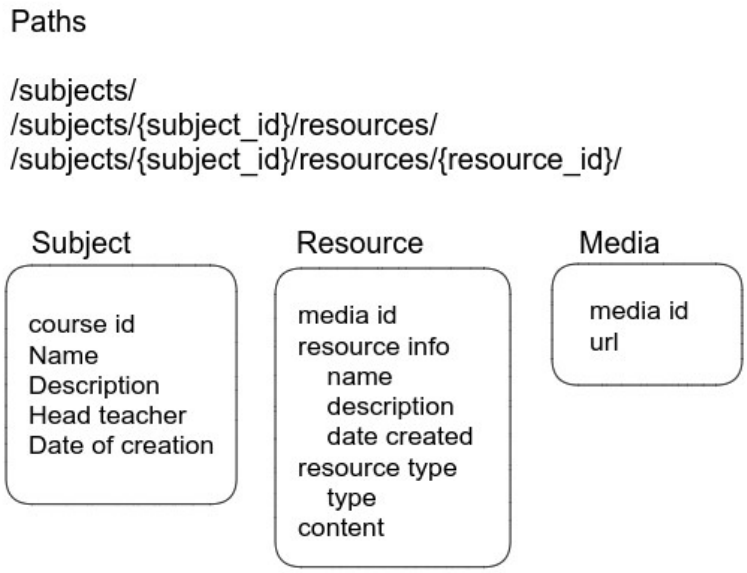

Figure 20. API resource and paths design

Tools

Now that the design is done, the API type must be chosen. The project's objective is to do the API calls using HTTP and stateless. The API has to follow CRUD over the subject and resource. CRUD stands for create, read, update and delete the specific resource. The project aims to allow the system to be implemented over different systems. The technology must have a set of standardized principles and communication between the client and the server. As developers, the programming language choice is essential. Hence, the type of API used in this project is a RESTful API that fulfils every requirement mentioned above, including the vast range of programming languages that supports this type.

The following natural step is choosing the programming language, framework or library to develop the REST API. Python will be the programming language used for the backend, and FastAPI will be the library for this project. Python helps us integrate AI models into the system straightforwardly with the help of different libraries. FastAPI is a Python library that lets developers create APIs with good performance, ease of use and intuitive API documentation generation. It has gained popularity in artificial intelligence as it is one of the fastest RESTful microservices.

The last step before starting the development is to choose where the information is stored. For this project, MongoDB is the one to choose to store the API resources. MongoDB is a NoSQL database that allows handling large amounts of data stored in custom documents

that are the fields of the resources. To host the database, Docker has been decided to use. Docker is a containerization platform that allows to deploy and manage it in a self-contained environment. Using it, we can manage the database's resources and make it reliable as it is isolated from host system updates.

Programming

When interacting with our MongoDB database from Python code, the pymongo library is used. Pymongo is a library that allows interaction with the MongoDB database. By importing it into our code, we can set up a mongo client with the information of the database, as can be seen in Figure 21. One parameter is the host that details what is introduced to the Docker IP and port of the database service. In this case, the Config class loads environmental variables to configure specific names, such as the database URL and name. This approach follows a practice of using environment variables that helps to manage different environments such as development, staging and production that may need different configuration without needing to change the code. Once the client is set up, a client object allows us to access the database through it. When the connection is made, the subjects collection and the media one are configured to have a specific id as the index. When creating a MongoDB, it has an _id, but we want a specific one, so it is created as an index to avoid repeating it. In case anything in this process fails, the service stops.

```
try:
    client = pymongo.MongoClient(Config.MONGO_URI)
    db = client[Config.MONGO_DB_NAME]
    db.subjects.create_index(
        [("course_id", pymongo.DESCENDING)], unique=True)
    db[Config.MEDIA_COLLECTION_NAME].create_index(
        [("media_id", pymongo.DESCENDING)], unique=True)
    print("Connection to MongoDB instance successful!")
except Exception as e:
    print("Error connecting to MongoDB instance:", e)
```

Figure 21. Database connection with Pymongo

The following step is to define the FastAPI application. In order to test and deploy the system, a server is required. Uvicorn is chosen as a server due to the support of async frameworks and is easy to use and configure. Once we have the necessary libraries installed and imported, the routes for the paths found in Figure 20 are defined. Each re-

source comprises dedicated endpoints, including GET, POST, PUT, and DELETE, specified in files with the corresponding resource name in the routers folder. These functions invoke other functions in the crud folder to interact with the database and execute the necessary tasks using the pre-defined models.

Models provide a way to manage data validation and serialization that, when reading or storing, are checked in order to keep consistency. The Pydantic library provides an easy way to declare models and schema for this task. When working with APIs or databases, Pydantic models define the expected data structure, ensuring that any incoming data meets specific requirements before processing. This characteristic not only helps to ensure the data's integrity but also helps debug any problems related to the development process, as a detailed error message is provided when the validation fails. Moreover, Pydantic models can generate documentation automatically based on the defined models. In addition, the library models support using examples to test endpoints that utilize the models for sending or receiving data, as seen in Figure 22.

```python
class SubjectCreate(BaseModel):
    course_id: str
    name: str
    description: str
    head_teacher: str

    class Config:
        schema_extra = {
            "example": {
                "course_id": "AT00BY24-3003",
                "name": "Programming",
                "description": "Learn basics of Python programming",
                "head_teacher": "Pepe",
            }
        }
```

Figure 22. Schema when creating a Subject

After defining the models, they can be added as responses of some endpoints to provide a way that makes the endpoint response consistent. The model is indicated on the endpoints of the router files. These methods that handle requests are usually backed by CRUD operations that interact with the database system. These operations are encapsulated in a separate module imported by the router. By doing this encapsulation, it is easier to debug in case of problems as each function has a specific task, and each resource of the API has its one.

Furthermore, the code of the router part is cleaner as it does not have any database query. The endpoint simply calls the appropriate function in the CRUD module that handles the details of the database interaction. In the case of the subjects, there are several functions where all subjects are obtained, one subject filtered by course_id, and the query to create a new subject, among others.

Finally, the transcription and question-answer functionality has to be implemented. In order to transcribe a video, the transcription process is triggered when a new media is created, providing the URL from a video. Initially, we attempted to upload video files, but providing them suitably for the frontend was not feasible due to how they were stored and managed. Therefore, the youtube-dlp library is used to download videos from Youtube. The Whisper model is then loaded, and for development purposes, it is set to "small", as available hardware cannot process the large model. Once the transcription process is complete, a JSON file consists of the Content model in Figure 23. It contains the whole text followed by a list of segments with information such as the segment's text, start time or end time that are useful for searching within the video. The last information given is the language of the video.

```
class Segment(BaseModel):
    id: int
    seek: int
    start: int
    end: int
    text: str
    tokens: List[int]
    temperature: int
    avg_logprob: int
    compression_ratio: int
    no_speech_prob: int


class Content(BaseModel):
    text: str
    segments: List[Segment] | None = None
    language: str | None = None
```

Figure 23. Pydantic Models that represent the transcription result

As for the question-answer functionality, it has its endpoint. Depending on whether the media_id is provided, it searches for the answer among all the resources of a subject or in a specific resource if the media_id is provided. The process of loading the question-answering model and performing inference is set up according to the example shown in Fig-

ure 19. Once the result is obtained, the answer is searched for in the different segments to provide the exact minute and second at which it was said. It is accomplished by dividing the query string into individual segments and then comparing each to the text in the segments. The segment with the most matches is considered the best match and returned as the answer. It is important because the answer may be spread across different segments, so this function helps identify the segment containing the answer.

Task queue

As the transcription process can take some time to complete, it was necessary to find a way to execute it asynchronously and avoid blocking overflowing the hardware resources. Moreover, there was a need to handle multiple requests for transcription without losing any of them and keep track of each task's progress and status. The solution was to use a task queue and a message broker. Celery, a distributed task queue that allows for the execution of tasks in the background and the management of task workflows, is the choice and Redis as message broker.

The transcription task was converted into an asynchronous Celery task to prevent blocking the API server and ensure efficient resource utilization. The Celery task queue is designed to handle every task related to transcription. Due to limited hardware capabilities, the queue is FIFO (First In, First Out), so only one worker can execute one task simultaneously. It is accomplished on the code by writing celery annotation and including a queue parameter containing the queue's name where that task is stored. The next task arrives at the end of the queue, and the worker consumes the first one when finished. This way, the task queue ensures that all transcription requests are processed in the order they are received and that no request is lost.

Similarly, the question-answering functionality was turned into a Celery task that receives the question and the media_id (if any) and returns the answer with the corresponding timestamp on the video. This task was put into a separate queue with several workers, as it does not require much computational power compared to transcription. Using multiple workers for the question-answering task enables concurrent execution and even load distribution, resulting in low response time and no long waiting periods for users to get results. Additionally, if every worker is doing a task and a new one is received, it is stored in the queue.

A message broker is needed to manage the task queue. It handles the communication between the Celery workers and the API. Redis is chosen as the message broker and is hosted on Docker in the same way as MongoDB. Redis allows storing and retrieving mes-

sages, which are used to send task requests and status updates between the application and the Celery workers. It was chosen due to the minimal configuration required to work.

The schema presented in Figure 24 illustrates the system's workflow. Whenever a request is received and requires the execution of a defined task, it is added to a queue in Redis based on the code's annotation. The message containing the request is stored in Redis regardless of whether Celery workers are online, making it reliable. Celery processes the message accordingly. In our case, the normal queue associated with the question-answer has several workers, and the task, when it is completed, the result can be consulted. The client retrieves the result using the API. In the case of the transcription task, the result is stored on the corresponding resource that the client can access through the get resource endpoint.

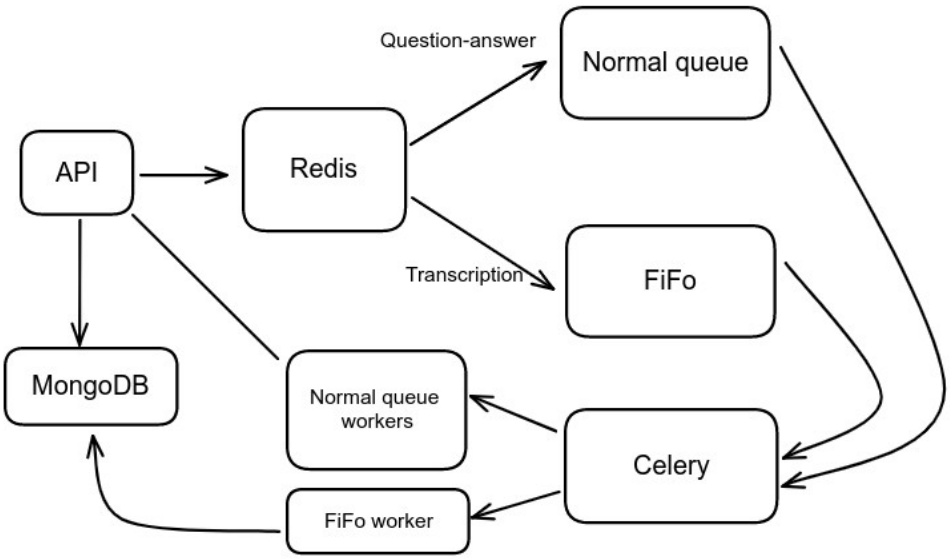

Figure 24. Schema of Task queue

Finally, Flower, a web interface designed expressly for Celery, was used to monitor the system's state, which included incoming tasks, queues, and the outcomes of processed tasks. It was used to check that the Celery system was running correctly and that any bugs or mistakes could be recognized and fixed immediately. Flower also enabled us to track Celery jobs in real time and quickly troubleshoot and optimize their performance. Flower allows to check the system's general health, monitor queue lengths, and examine

individual worker performance, making it a perfect tool for development and administration.

API Documentation

One key point of why FastAPI was chosen is because it provides two types of automatic API documentation, Swagger and ReDoc. Swagger allows interaction with the API and tries out different requests and responses. ReDoc is another documentation simpler than Swagger and with a more user-friendly interface. Both provide a visual representation of the API endpoints, their corresponding parameters to make a request, and any responses that may be turned in. This visual representation can be seen in Figure 25.

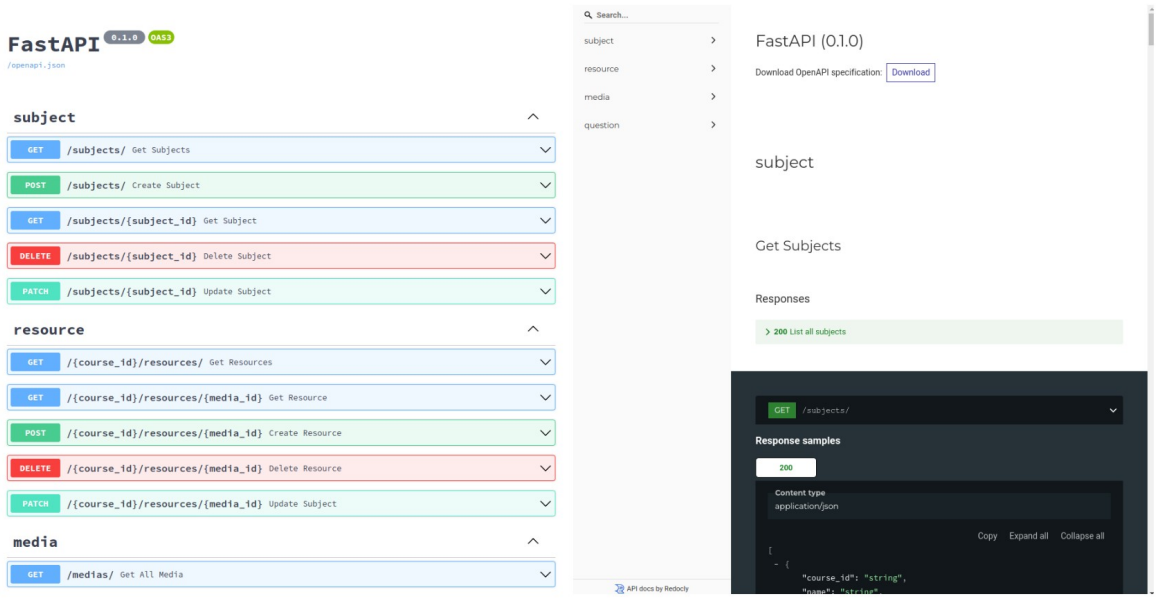

Figure 25. Swagger and ReDocs API documentation

Another key feature of FastAPI is its use of Pydantic, which automatically generates API sachems. The generation is based on the Python data models defined in the code. It eliminates the manual part of the documentation, reducing time and the risk of errors.

7.5 User interface

Once the API has been developed and tested, creating a user-friendly demo showcase interface for the application is essential. At this point, is where the frontend development becomes significant. For this final thesis, a web-based application has been created that ensures a consistent user experience across various operating systems. It is worth noting

that the web application is not responsive and is designed for desktop use only, as this frontend is only a demo to show what the API can do. The frontend application is specifically designed to interact with the API endpoints, providing users with a convenient and intuitive way to access the application's features.

Figure 26 shows a wireframe designed using Penpot, an open-source web application for creating wireframes, mockups, and prototypes. The wireframe provides a visual representation of the proposed design for the frontend interface. The interface is designed to be simple and consistent, with a CSS framework ensuring a uniform design across all pages. At this stage, a mockup or prototype was not deemed necessary and therefore was not developed.

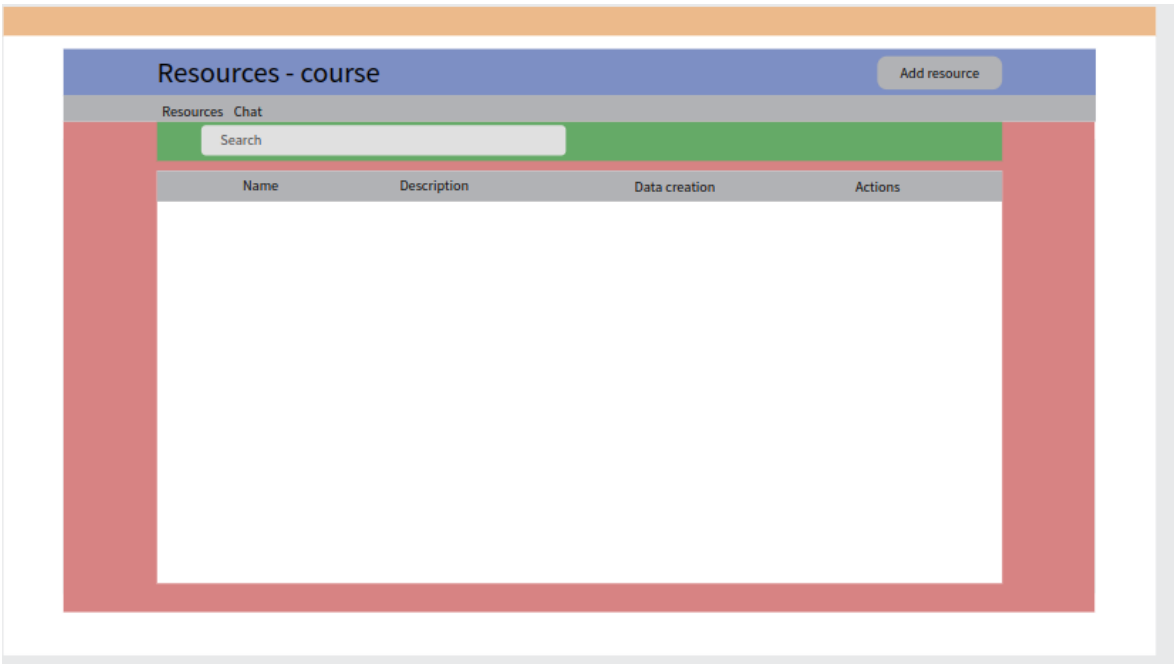

Figure 26. Wireframe design of resources view

The Vite development server was used for this project to deploy the frontend web application. Vite is a build tool chosen for its fast and efficient development experience. It supports hot module replacement that allows saving the file that has been edited and instantly seeing the modification on the website, making the development process quicker and more efficient.

Regarding programming language, TypeScript was chosen and React as a library. TypeScript is a derived language of JavaScript that provides a strong typing system, making it easier to detect and prevent common errors. React library was selected as one of the most popular libraries alongside Vue. It provided features to build dynamic and interactive user interfaces more quickly, one of the main key points of why React was chosen.

Regarding styling, the Bootstrap CSS framework was utilized to guarantee a consistent design throughout the entire application. This framework provides various styles that facilitate customization by adding the library's classes into the HTML tags. While most of the components were created from scratch for learning, the application also used the react-toastify library and react-bootstrap tabs components, simplifying the implementation of specific features.

The frontend of the demo showcase application consists of three main views: subjects, resources, and video. In the subjects view, users can create new subjects by clicking the "Create Subject" button, which opens a modal with four input fields: course ID, name, description, and head teacher. When clicking the "Save" button, the frontend makes a fetch request to the API endpoint, creating a new subject. The list of subjects is displayed in a table with information such as name, description, head teacher and an action column with several buttons. Users can move to the resources view of a subject by clicking the "Access" button and modify the subject information by clicking the "Modify" button. The last button is "Delete". By clicking, it displays a confirmation modal that deletes the subject. The table also allows users to sort by each field of the header except for the action column, and there is also a filter input box that can filter by any text found in any field of the table.

In the resources view, users can see a table listing every resource in a subject. It displays their name, description, and creation date. As the subject table, users can sort it by an input box filter or by each column of the header. At the top of the view, the subject's name, description, and head teacher are displayed, along with a button to create resources. An example is shown in Figure 27. When users click on this button, a modal opens that accepts the video's name, description, and URL. As the subjects view, the frontend makes a POST fetch request to the API endpoint. Unlike the subjects view, the resources view has a tab system that includes two tabs: "Resources" and "Questions." In the "Questions" tab, users can ask questions about any of the videos' transcriptions in the list. If the resource is newly created, it may not have a transcription yet, as it is still processing. When a user submits a question, a celery task is created in the backend, and the frontend makes several fetch requests until the answer is available. In addition to the answer, a button with the video's timestamp is displayed, which leads to the video view.

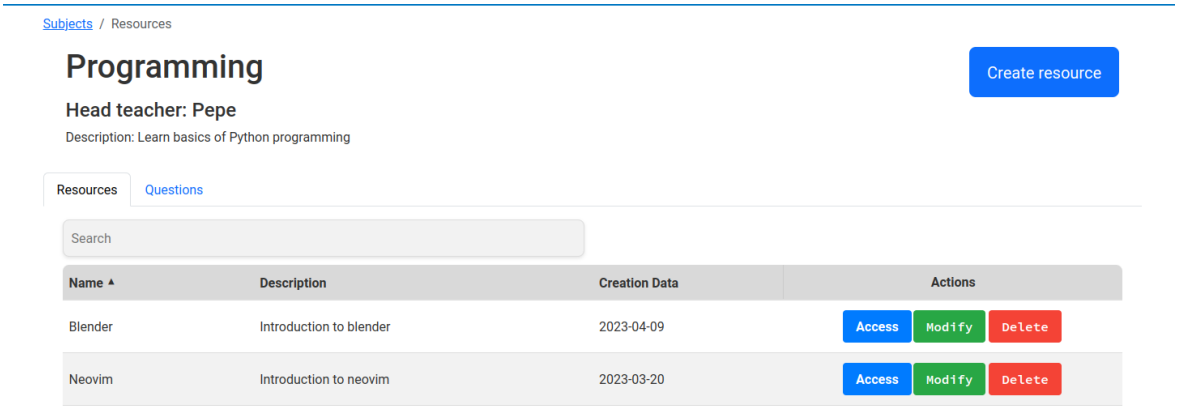

Figure 27. Resource view

The last view is the video that includes the visualization of the video using the YouTube player. Initially, there were problems when using an old API endpoint that allowed users to upload their video files to the API. While the backend process worked fine, the video player only worked well on Firefox. In other browsers, the video would not display or could not change the timestamp of the video player. This problem is the consequence of why the API and backend process was rewritten to use YouTube player. As the resources view, the video one also has tabs. In this case, there are four tabs: "Video," "Questions," "Segments," and "Transcription." The "Video" tab contains the video player and a "Current Segment" display showing the segment of the video's timestamp changing while the video plays. The "Questions" tab is the same as the one on the resources view, but the only difference is that users can only ask questions related to the selected resource. It also includes a button to set the timestamp of the answer and changes the tab, as seen in Figure 28. The "Segments" tab contains a list of buttons with the text of all the segments that are in the video. This tab allows users to search for a specific segment, and when they click on the button, it opens the video at the segment's timestamp. Finally, the "Transcription" tab contains the complete transcription text of the video.

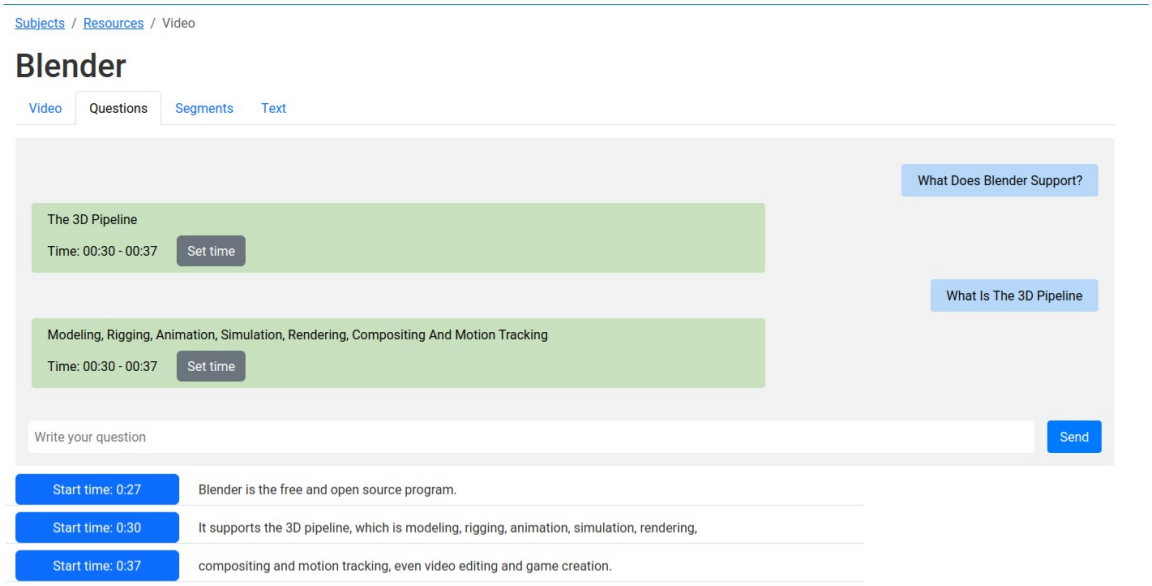

Figure 28. Answer to the question and the segments to check

In conclusion, the frontend web application, with its various views, has provided users with an efficient and user-friendly way to interact with the API. With the use of TypeScript and React, along with the Vite server for deployment, the application has been designed quickly and used to learn those technologies. The different views have been carefully designed to allow users to easily create, modify, and delete subjects and resources, as well as to ask questions and navigate through video segments. While the application is only a demo interface, it is a powerful tool to showcase the capabilities of the underlying API to non-technical individuals.

8 Conclusions

During this thesis, the focus has been on exploring the potential of artificial intelligence in transforming education and its various applications. The theoretical aspect of the study discussed the history of AI, its different types, and how they have evolved. The thesis also emphasised the importance of natural language processing in language understanding and how it can contribute to education. Finally, it introduced the concept of REST API and its significance in building new applications over it.

The main objective of the practical case was to create a REST API that provides the ability to solve the doubts about the explanation in online classes without the need to watch the full video. The goal was achieved by combining two different open-source artificial intelligence models. The first model, Whisper AI, was used to transcribe the audio content of a video, while the second model, Roberta-base-squad2, was used to answer questions related to the video's content.

During the development, several problems were encountered. The most significant challenge was the limited hardware resources on the development side. Due to the vast amount of video RAM required for some artificial intelligence processes, the smallest available model was used for the transcription process. Furthermore, as the API needed to handle multiple calls simultaneously, some of which required the execution of AI processes, a task queue was implemented to streamline the AI process. Another issue was the difficulty of receiving a video uploaded to the API. The video reproduction did not function properly in every browser, and changing the timestamp was not always possible.

The final results of the research and development of a REST API for automatic transcription and question-answering using open-source models were successful. The study's main objective was to explore educational learning methods and provide a solution that facilitates the learning process. The implementation of the API accomplished this objective, allowing for the automatic transcription of audio content and providing accurate answers to related questions. Different tests were conducted on the API, and the results showed that it is effective in extracting extractive question answering.

Regarding the results, there is potential for the project to be enhanced by using new generative models such as Llama and Alpaca and their many variants. Further development on the interaction with AI calls could allow these models to extract more information from the transcription and generate detailed and longer answers, thereby improving the learning process. Including new models like GPT4ALL, a derivative of Llama, could also revolutionise the question-answering capabilities of the API by enabling it to hold conversations.

For those who prefer closed-source models, ChatGPT could be an excellent option for the project's development while providing video transcription as context.

Overall, developing this thesis provides me with valuable experience and knowledge of various programming languages such as Python and TypeScript. I also learned to use various technologies, such as Fastapi, task queues, and some aspects of React development. As I am deeply interested in AI, developing this practical case has been a difficult job but motivating me to be even more interested in artificial intelligence and its applications.

References

Achary, N. 2019. Artificial Intelligence to transform 10 Industries. Medium. Retrieved 27 February 2023. Available at https://becominghuman.ai/artificial-intelligence-to-transform-10-industries-498338359f415

Actix. Website. Retrieved on 8 March 2023. Available at https://actix.rs/

Adservio. 2022. What is Flutter and its advantages? Retrieved on 5 March 2023. Available at https://www.adservio.fr/post/what-is-flutter-and-what-are-its-advantages

Allen. J. 2022. Théâtre D'opéra Spatial. Retrieved on 3 March 2023. Available at https://static01.nyt.com/images/2022/09/01/business/00roose-1/merlin_212276709_3104aef5-3dc4-4288-bb44-9e5624db0b37-jumbo.jpg?quality=75&auto=webp

Angular. 2022. What is Angular? Retrieved on 5 March 2023. Available at https://angular.io/guide/what-is-angular

Anyoha, R. 2017. The History of Artificial Intelligence. Article. Harvard. Accessed on 24 October 2022. Available at https://sitn.hms.harvard.edu/flash/2017/history-artificial-intelligence/

Appshopper. 2021. A Quick Guide on Mobile App Wireframe vs Mockup vs Prototype. Retrieved on 5 March 2023. Available at https://www.appschopper.com/blog/quick-guide-on-mobile-app-wireframe-vs-mockup-vs-prototype/

ARTiBA. 2021. How AI is Transforming the Education Industry. Retrieved on 28 February 2023. Available at https://www.artiba.org/blog/how-ai-is-transforming-the-education-industry

AWS. a. What is overfitting? Retrieved on 1 March 2023. Available at https://aws.amazon.com/what-is/overfitting/

AWS. b. What is a RESTful API? Retrieved on 8 March 2023. Available at https://aws.amazon.com/what-is/restful-api/

Babich, N. 2019a. Top Website Layouts That Never Grow Old. Retrieved on 5 March 2023. Available at https://xd.adobe.com/ideas/principles/web-design/11-website-layouts-that-made-content-shine-in-2019/

Babich, N. 2019b. User Centered Design Principles & Methods. Retrieved on 5 March 2023. Available at https://xd.adobe.com/ideas/principles/human-computer-interaction/user-centered-design/

Baheti, P. 2023. A newbie-Friendly Guide to Transfer Learning. Retrieved on 2 March 2023. Available at https://www.v7labs.com/blog/transfer-learning-guide#h1

Berkeley Extension. a. Top 6 AI Programming Languages to Learn in 2023. Retrieved on 3 March 2023. Available at https://bootcamp.berkeley.edu/blog/ai-programming-languages/#:~:text=AI%20Programming%20With%20C%2B%2B&text=It%20executes%20code%20quickly%2C%20making,programs%20that%20run%20exceptionally%20well.

Berkeley Extension. b. What Does a Front End Web Developer Do? Retrieved on 5 March 2023. Available at https://bootcamp.berkeley.edu/resources/coding/learn-web-development/what-does-a-front-end-web-developer-do/

Bilal, A. 2022. Best practices for REST API Security. Retrieved on 8 March 2023. Available at https://rapidapi.com/guides/practices-rest-security

Billion Acts. Education and Community Development. Retrieved on 27 February 2023. Available at https://www.billionacts.org/focus-area/education-and-community-development

Brown, S. 2021. Machine learning, explained. Retrieved on 25 October 2022. Available at https://mitsloan.mit.edu/ideas-made-to-matter/machine-learning-explained

Cambridge Dictionary. Mock-up. Retrieved on 5 March 2023. Available at https://dictionary.cambridge.org/dictionary/english/mock-up

Celery. Introduction to Celery. Retrieved on 8 March 2023. Available at https://docs.celeryq.dev/en/stable/getting-started/introduction.html

Christensson, P. 2020. Frontend. Retrieved on 5 March 2023. Available at https://techterms.com/definition/frontend

CIS International School. 2019. The role of education in social development. Retrieved on 27 February 2023. Available at https://cisedu.com/en-gb/world-of-cis/news/the_role_of_education_in_social_development_ccfs21/

Civati, A. 2021. A.I. in the Education Industry - A deep and global change of view. Retrieved on 28 February 2023. Available at https://www.linkedin.com/pulse/ai-education-industry-deep-global-change-view-alessandro-civati?trk=articles_directory

Computer Hope. 2022. Server. Retrieved on 7 March 2023. Available at https://www.computerhope.com/jargon/s/server.htm

Council of Europe portal. 2015. Knowledge, skills, attitudes and values supporting human rights education. Retrieved on 27 February 2023. Available at https://www.coe.int/en/

web/gender-matters/knowledge-skills-attitudes-and-values-supporting-human-rights-education#

Cristina, S. 2022. The Transformer Model. Retrieved on 2 March 2023.

DeepMind. AlphaGo. Retrieved on 24 October 2022. Available at https://www.deepmind.com/research/highlighted-research/alphago

Devlin, J. & Chang, M. & Lee, K. & Toutanova, K. 2019. BERT: Pre-trained of Deep Bidirectional Transformers for Language Understanding. Retrieved on 12 March 2023. Available at https://arxiv.org/pdf/1810.04805.pdf

Dhanushka, D. 2020. The stuff that every developer should know about message queues. Retrieved on 8 March 2023. Available at https://medium.com/event-driven-utopia/the-stuff-that-every-developer-should-know-about-message-queues-a9452ac9c9d

Dhingra, S. 2021. Simplified Mathematics behind Neural Networks. Retrieved on 2 March 2023. Available at https://towardsdatascience.com/simplified-mathematics-behind-neural-networks-f2b7298f86a4

Dictionary. Markup. Retrieved on 5 March 2023. Available at https://www.dictionary.com/browse/markup

Dilmegani, C. 2020. Bias in AI: What it is, Types, Examples & 6 Ways to Fix it in 2023. Retrieved on 1 March 2023. Available on https://research.aimultiple.com/ai-bias/

Dilmegani, C. 2023. Large Language Model Examples in 2023. In AIMultiple. Retrieved on 11 March 2023. Available at https://research.aimultiple.com/large-language-models-examples/

Doe, J. 2020. Rule Based Systems. Article. ProfessionalAI. Accessed on 28 October 2022. Available at https://www.professional-ai.com/rule-based-systems.html

Duolingo. 2022. What is Duolingo?. Retrieved on 28 February 2023. Available at https://support.duolingo.com/hc/en-us/articles/204829090-What-is-Duolingo-

Educause. 2022. 3 Ways AI Can Help Students with Disabilities. Available on 28 February 2023. Available at https://er.educause.edu/articles/2022/6/3-ways-ai-can-help-students-with-disabilities

FastAPI. Website. Retrieved on 8 March 2023. Available at https://fastapi.tiangolo.com/

Fawcett, A. 2021. Introuction to Human-Computer Interaction & Design Principles. Retrieved on 5 March 2023. Available at https://www.educative.io/blog/intro-human-computer-interaction

Flower. Github repository. Retrieved on 8 March 2023. Available at https://github.com/mher/flower

Flutter. Website. Retrieved on 5 March 2023. Available at https://flutter.dev/

Frankenfield, J. 2022. Artificial Intelligence: What It Is and How It Is Used. Retrieved on 27 February 2023. Available at https://www.investopedia.com/terms/a/artificial-intelligence-ai.asp

Fulton, J. 2018. Emulating Logical Gates with a Neural Network. Structure of a neuron. Retrieved on 2 March 2023. Available at https://towardsdatascience.com/emulating-logical-gates-with-a-neural-network-75c229ec4cc9

Garcia, E. 3 Best Practices for API documentation. Retrieved on 8 March 2023. Available at https://www.pandium.com/blogs/3-best-practices-for-api-documentation

GCF Global. Computer Basics: Understanding Applications. Retrieved on 5 March 2023. Available at https://edu.gcfglobal.org/en/computerbasics/understanding-applications/1/

GeeksForGeeks. 2020. Difference between Spreadsheet and Database. Retrieved on 9 March 2023. Available at https://www.geeksforgeeks.org/difference-between-spreadsheet-and-database/

GeeksForGeeks. 2022. Difference between SQL and NoSQL. Retrieved on 9 March 2023. Available at https://www.geeksforgeeks.org/difference-between-sql-and-nosql/

GeeksForGeeks. 2023. Reinforcement Learning. Retrieved on 1 March 2023. Available at https://www.geeksforgeeks.org/what-is-reinforcement-learning/

Ghiya, V. 2022. What are the Natural Language Processing Challenges, and How to Fix them?. In ThinkML. Retrieved on 10 March 2023. Available at https://thinkml.ai/what-are-the-natural-language-processing-challenges-and-how-to-fix-them/

Google Calc. 2022. Whisper Perf- Public 2022-09-27. Google docs. Retrieved on 28 March 2023. Available at https://docs.google.com/spreadsheets/d/1xdaK-RJZ2ft-MKBME45aAeEmMHSJSxb3wW8-GzT1whgg/edit#gid=1647983587

Google. 2022. Descending into ML: Training and Loss. Retrieved on 1 March 2023. Available at https://developers.google.com/machine-learning/crash-course/descending-into-ml/

training-and-loss#:~:text=Training%20a%20model%20simply%20means,is%20called%20empirical%20risk%20minimization.

Gülen, K. 2023. How AI improves education with personalized learning at scale and other new capabilities. Retrieved on 27 February 2023. Available at https://dataconomy.com/2023/02/artificial-intelligence-in-education/

Hamilton, T. 2023. GUI Testing - UI Test Cases (Examples). Retrieved on 5 March 2023. Available at https://www.guru99.com/gui-testing.html

Herbert, D. 2022. What is React.js? (Uses, Examples, & More). Retrieved on 6 March 2023. Available at https://blog.hubspot.com/website/react-js

Hern, A. 2016. AlphaGo: its creator on the computer that learns by thinking. The Guardian. Retrieved on 22 February 2023. Available at https://www.theguardian.com/technology/2016/mar/15/alphago-what-does-google-advanced-software-go-next

Hufford, B. 2022. What is a Mockup? (+How to Create a Mockup in 2022). Retrieved on 5 March 2023. Available at https://cliquestudios.com/mockups/

Huggingface. a. Question Answering. Retrieved on 15 May 2023. Available at https://huggingface.co/tasks/question-answering

Huggingface. b. Transformers. Github repository. Retrieved on 3 March 2023. Available at https://github.com/huggingface/transformers

IBM. a. What is artificial intelligence (AI)?. IBM. Retrieved on 22 February 2023. Available at https://www.ibm.com/topics/artificial-intelligence

IBM. b. Deep Blue. Post. IBM. Retrieved on 24 October 2022. Available at https://www.ibm.com/ibm/history/ibm100/us/en/icons/deepblue/

IBM. c. What is supervised learning?. Retrieved on 1 March 2023. Available at https://www.ibm.com/topics/supervised-learning#:~:text=Supervised%20learning%2C%20also%20known%20as,data%20or%20predict%20outcomes%20accurately

IBM. d. What is unsupervised learning?. Retrieved on 1 March 2023. Available at https://www.ibm.com/cloud/learn/unsupervised-learning#:~:text=Unsupervised%20learning%2C%20also%20known%20as,the%20need%20for%20human%20intervention.

IBM. e. What is a neural network?. Retrieved on 2 March 2023. Available at https://www.ibm.com/topics/neural-networks

IBM. f. What is a REST API? Retrieved on 8 March 2023. Available at https://www.ibm.-com/topics/rest-apis

IBM. g. What are message brokers? Retrieved on 8 March 2023. Available at https:// www.ibm.com/topics/message-brokers

Ijaz, A. 2022. Native UI: Future of App Development. Retrieved on 5 March 2023. Limited availability at https://medium.com/illumination/native-ui-future-of-app-development-e96a85ef95ed

Indeed Editorial Team. 2023. Types of Computers Servers and How They Function. Retrieved on 7 March 2023. Available at https://www.indeed.com/career-advice/career-development/types-of-servers

Indeed. 2022. Front End Developer Job Description: Top Duties and Qualifications. Retrieved on 5 March 2023. Available at https://www.indeed.com/hire/job-description/front-end-developer#:~:text=A%20Front%20End%20Developer's%20main,and%20design %20of%20a%20website

Ingalls, S. 2021. What is a server and what do servers do? Retrieved on 7 March 2023. Available at https://www.serverwatch.com/guides/what-is-a-server/

Interaction Design. a. What is Human-Computer Interaction (HCI)? Retrieved on 5 March 2023. Available at https://www.interaction-design.org/literature/topics/human-computer-interaction

Interaction Design. b. What are Design Guidelines? Retrieved on 5 March 2023. Available at https://www.interaction-design.org/literature/topics/design-guidelines

Jajal, T. D. 2018. Distinguishing between Narrow AI, General AI and Super AI. Medium. Retrieved on 1 November 2022. Available at https://medium.com/mapping-out-2050/distinguishing-between-narrow-ai-general-ai-and-super-ai-a4bc44172e22

JavaTPoint. Types of Artificial Intelligence - Javatpoint. Article. Retrieved on 1 November 2022. Available at https://www.javatpoint.com/types-of-artificial-intelligence

Julia. The Julia Programming Language. Retrieved on 3 March 2023. Available at https:// julialang.org/

Jupyter. Website. Retrieved on 3 March 2023. Available at https://jupyter.org/

Juviler, J. 2021. 9 Guidelines & Best Practices for Exceptional Web Design and Usability. Retrieved on 5 March 2023. Available at https://blog.hubspot.com/blog/tabid/6307/bid/ 30557/6-guidelines-for-exceptional-website-design-and-usability.aspx

Kapronczay, M. 2022. A Beginner's Guide to Language Models. In Builtin. Retrieved on 11 March 2023. Available at https://builtin.com/data-science/beginners-guide-language-models

Keras. Website. Retrieved on 3 March 2023. Available at https://www.tensorflow.org/

Khan, S. 2019. BERT, RoBERTa, DistilBERT, XLNet — which one to use? In Medium. Retrieved on 12 March 2023. Available at https://towardsdatascience.com/bert-roberta-distilbert-xlnet-which-one-to-use-3d5ab82ba5f8

Khurana, D. & Koli, A. & Khatter, K. & Singh, S. 2022. Natural language processing: state of the art, current trends and challenges. Multimed Tools Appl 82, 3713–3744 (2023). Retrieved on 10 March 2023. Available at DOI https://doi.org/10.1007/s11042-022-13428-4

Kimbarovsky, R. 2023. Branding: What it is, Why it's important, How to Build a Strong Brand, and Examples. Retrieved on 5 March 2023. Available at https://www.crowdspring.com/blog/branding/

Kotlin. 2023. Cross-platform mobile development. Retrieved on 5 March 2023. Available at https://kotlinlang.org/docs/cross-platform-mobile-development.html

Kumar, V. 2019. Python Vs R: What's Best for Machine Learning. Retrieved on 3 March 2023. Available at https://towardsdatascience.com/python-vs-r-whats-best-for-machine-learning-93432084b480

Lee, A. 2019. Why NLP is important and it'll be the future - our future. In Medium. Retrieved on 10 March 2023. Limited availability at https://towardsdatascience.com/why-nlp-is-important-and-itll-be-the-future-our-future-59d7b1600dda

Lee, A. 2023. What Are Large Language Models Used For? In Nvidia. Retrieved on 11 March 2023. Available at https://blogs.nvidia.com/blog/2023/01/26/what-are-large-language-models-used-for/

Lee, S. & Ju, E. & Woo Choi, S. & Lee, H. & Bo Shim, J. & Hwan Chang, K. & Hyeon Kim, K. & Yong Kim, C. 2019. Prediction of Cancer Patient Outcomes Based on Artificial Intelligence. Artificial Intelligence - Scope and Limitations. Retrieved on 21 February 2023. Available at DOI 10.5772/intechopen.81872

Locatelli, R. 2018. Education as a public and common good: reframing the governance of education in a changing context. UNESCO. Retrieved on 27 February 2023. Available at https://unesdoc.unesco.org/ark:/48223/pf0000261614?posInSet=5&queryId=N-EXPLORE-aaee6eff-d075-4efa-b76c-feac9a553fa3

MacFarlane, A. 2013. Information, Knowledge & Intelligence. Retrieved on 21 February 2023. Available at https://philosophynow.org/issues/98/Information_Knowledge_and_Intelligence

Mahajan, A. 2020. What you should choose - Native vs. Cross Platform? Retrieved on 5 March 2023. Available at https://www.sphinx-solution.com/blog/what-you-should-choose-native-vs-cross-platform/

Manchanda, A. 2022. The Ultimate Guide to Cross Platform App Development Frameworks in 2023. Retrieved on 5 March 2023. Available at https://www.netsolutions.com/insights/cross-platform-app-frameworks-in-2019/

Marijan, B. 2022. Server Operating Systems: Server OS Types & How to Choose. Retrieved on 7 March 2023. Available at https://phoenixnap.com/kb/server-operating-system

Marr. B. 2023. GPT-4 Is Coming – What We Know So Far. Retrieved on 3 March 2023. Available at https://www.forbes.com/sites/bernardmarr/2023/02/24/gpt-4-is-coming--what-we-know-so-far/

Marsden, P. 2017. Artificial Intelligence Timeline Infographic – From Eliza to Tay and beyond. Retrieved on 25 October 2022. Available at https://digitalwellbeing.org/artificial-intelligence-timeline-infographic-from-eliza-to-tay-and-beyond/

Martin, M. 2023. Difference between Website and Web Application (Web App). Retrieved on 5 March 2023. Available at https://www.guru99.com/difference-web-application-website.html#2

Masuyama, Y. & Chang, X. & Cornell, S. & Watanabe, S. & Ono, N. 2022 End-to-End Integration of Speech Recognition, Dereverberation, Beamforming, and Self-supervised Learning Representation. Retrieved on 12 March 2023. Available at https://arxiv.org/abs/2210.10742

McCarthy, J. 2007. What is artificial Intelligence?. Computer Science Department Stanford University. Retrieved on 27 January 2023. Available at https://www-formal.stanford.edu/jmc/whatisai.pdf

McFarland, A. 2022. 10 Best AI Tools for Education. Retrieved on 3 March 2023. Available at https://www.unite.ai/10-best-ai-tools-for-education/

Merritt, R. 2022. What is a transformer model?. Retrieved on 2 March 2023. Available at https://blogs.nvidia.com/blog/2022/03/25/what-is-a-transformer-model/

Microsoft. 2022. Windows Server. Retrieved on 7 March 2023. Available at https://learn.-microsoft.com/en-us/windows/win32/srvnodes/windows-server

Mishra, P. 2021. Understanding Masked Language Models (MLM) and Causal Language Models (CLM) in NLP. In Medium. Retrieved on 12 March 2023. Limited availability at ht-tps://towardsdatascience.com/understanding-masked-language-models-mlm-and-causal-language-models-clm-in-nlp-194c15f56a5

MongoDB. What is NoSQL? Retrieved on 9 March 2023. Available at https://www.mon-godb.com/nosql-explained

Mozilla. 2023. What is JavaScript? Retrieved on 5 March 2023. Available at https://de-veloper.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript

Nagyfi, R. 2018. The differences between Artificial and Biological Neural Networks. Re-trieved on 2 March 2023. Available at https://towardsdatascience.com/the-differences-between-artificial-and-biological-neural-networks-a8b46db828b7

NestJs. Website. Retrieved on 8 March 2023. Available at https://nestjs.com/

Nguyen, V. 2022. A Guide to Understanding Essental Speech AI Terms. In Nvidia De-veloper. Retrieved on 12 March 2023. Available at https://developer.nvidia.com/blog/a-guide-to-understanding-essential-speech-ai-terms/

Numenta. 2022. AI is harming our planet: addressing AI's staggering energy cost. Re-trieved on 2 March 2023. Available at https://www.numenta.com/blog/2022/05/24/ai-is-harming-our-planet/

Nvidia. Pretrained AI Models. Retrieved on 2 March 2023. Available at https://developer.n-vidia.com/ai-models

Nyakundi, H. 2022. API testing best practices - How to Test APIs for Beginners. Retrieved on 8 March 2023. Available at https://www.freecodecamp.org/news/rules-of-api-testing-for-beginners/

Onose, E. 2023. Explainability and Auditability in ML: Definitions, Techniques, and Tools. Retrieved on 1 March 2023. Available at https://neptune.ai/blog/explainability-auditability-ml-definitions-techniques-tools

OpenAI. 2022. Introducing ChatGPT. Retrieved on 3 March 2023. Available at https://openai.com/blog/chatgpt

Oracle. What is a Database? Retrieved on 9 March 2023. Available at https://www.oracle.-com/database/what-is-database/

Pajak, B. & Bicknell, K. At Duolingo, humans and AI work together to create a high-quality learning experience. Duolingo. Retrieved on 28 February 2023. Available at https://blog.-duolingo.com/how-duolingo-experts-work-with-ai/

Peterson, R. 2023. Database Design in DBMS Tutorial: Learn Data Modeling. Retrieved on March 9 2023. Available at https://www.guru99.com/database-design.html

PHP. What is PHP? Retrieved on 8 March 2023. Available at https://www.php.net/manual/en/intro-whatis.php

Python. What is Python? Executive Summary. Retrieved on 3 March 2023. Available at https://www.python.org/doc/essays/blurb/

Pytorch. Website. Retrieved on 3 March 2023. Available at https://pytorch.org/

RabbitMQ. Website. Retrieved on 9 March 2023. Available at https://www.rabbitmq.com/

Rainergewalt. 2021. Supervised vs Unsupervised vs Reinforcement Learning. Retrieved on 2 March 2023. Available at https://starship-knowledge.com/supervised-vs-unsupervised-vs-reinforcement

React. Website. Retrieved on 6 March 2023. Available at https://reactjs.org/

Red Hat. 2020. What is a REST API? Retrieved on 8 March 2023. Available at https://www.redhat.com/en/topics/api/what-is-a-rest-api

Redis. Website. Retrieved on 9 March 2023. Available at https://redis.io/

Rella, S. 2022. Essential Guide to Automatic Speech Recognition Technology. In Nvidia Developer. Retrieved on 12 March 2023. Available at https://developer.nvidia.com/blog/essential-guide-to-automatic-speech-recognition-technology/

ReQtest. Why is the difference between functional and Non-functional requirements important? Retrieved on 5 March 2023. Available at https://reqtest.com/requirements-blog/functional-vs-non-functional-requirements/

Roose Kevin. 2022. An A.I.-Generated Picture Won an Art Prize. Artists Aren't Happy. Retrieved on 3 March 2023. Limited availability at https://www.nytimes.com/2022/09/02/technology/ai-artificial-intelligence-artists.html

Russell, S. & Norvig, P. 2009. Artificial Intelligence A modern Approach. . Third Edition Global. Pearson Series in Artificial Intelligence. United Kingdom: Pearson.

Russell, S. & Norvig, P. 2021. Artificial Intelligence A modern Approach. . Fourth Edition Global. Pearson Series in Artificial Intelligence. United Kingdom: Pearson.

Rust. Website. Retrieved on 8 March 2023. Available at https://www.rust-lang.org/

Ryabtsev, A. 2022. 8 Reasons Why Python is Good for AI and ML. Retrieved on 3 March 2023. Available at https://djangostars.com/blog/why-python-is-good-for-artificial-intelligence-and-machine-learning/

Samuel, A. 1959. Some Studies in Machine Learning Using the Game of Checkers. IBM Journal of Research and Development, vol. 3, no. 3, pp. 210-229. Retrieved on 1 March 2023. Available at DOI 10.1147/rd.33.0210.

Sarkar, D. 2018. A Comprehensive Hands-on Guide to Transfer Learning with Real-World Applications in Deep Learning. Retrieved on 2 March 2023. Available at https://towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27aA

Scylladb. Differences between SQL Database and NoSQL Database. Retrieved on 9 March 2023. Available at https://www.scylladb.com/learn/nosql/nosql-vs-sql/

Secret Society Software. CLIPS: A Tool for Building Expert Systems. Retrieved on 15 January 2023. Available at https://clipsrules.net/

Seo, K. & Tang, J. & Roll, I. & Fels, S. & Yoon, D. 2021. Storyboard example of Adaptive Quiz. Retrieved on 28 February 2023. Available at https://educationaltechnologyjournal.springeropen.com/articles/10.1186/s41239-021-00292-9/figures/1

Seobility. REST API. Retrieved on 8 March 2023. Available at https://www.seobility.net/en/wiki/REST_API

Seon. Blackbox Machine Learning. Retrieved on 1 March 2023. Available at https://seon.io/resources/dictionary/blackbox-machine-learning/#:~:text=In%20general%20terms%2C%20blackbox%20machine,of%20transparency%20in%20this%20technology.

Shao, C. 2019. Approach pre-trained deep learning models with caution. Retrieved on 2 March 2023. Available at https://medium.com/comet-ml/approach-pre-trained-deep-learning-models-with-caution-9f0ff739010c

Sharna, K. 2022. What is the difference between knowledge and education?. Retrieved on 27 February 2023. Available at https://www.theasianschool.net/blog/what-is-the-difference-between-knowledge-and-education/#2-what-is-education

Sketch. 2022. Wireframe vs mockup vs prototype: What's the difference? Retrieved on 5 March 2023. Available at https://www.sketch.com/blog/wireframe-vs-mockup-vs-prototype/

StarDust Testing. Hybrid Apps: An overview of advantages, limitations & consequences for your testing phases. Retrieved on 4 November 2022. Available at https://www2.star-dust-testing.com/en/blog-en/hybrid-apps

Stefanus, R. 2019. Conventional Programming vs Machine Learning. Medium. Retrieved on 1 March 2023. Available at https://rstefanus16.medium.com/conventional-program-ming-vs-machine-learning-a3b7b3425531

Subhashana, H. 2021. Introduction to Message Brokers. Retrieved on 8 March 2023. Available at https://hasithas.medium.com/introduction-to-message-brokers-c4177d2a9fe3

Taylor, D. 2023. What is Data Modelling? Types (Conceptual, Logical, Physical). Retrieved on March 9 2023. Available at https://www.guru99.com/data-modelling-conceptual-logical.html#6

TensorFlow. Website. Retrieved on 3 March 2023. Available at https://www.tensor-flow.org/

The R Foundation. What is R? Retrieved on 3 March 2023. Available at https://www.r-pro-ject.org/about.html

Tibco. What is a Neural Network?. Structure of neural network. Retrieved on 2 March 2023. Available at https://www.tibco.com/reference-center/what-is-a-neural-network

Tilbe, A. 2022. Why Do So Many People Struggle With Understanding Artificial Intelli-gence?. Retrieved on 22 February 2023. Limited availability at https://uxplanet.org/why-do-so-many-people-struggle-with-understanding-artificial-intelligence-9dca76e1b6f9

Toczyska, K. L. How to engage your clients in the design process. Retrieved on 5 March 2023. Available at https://uxdesign.cc/how-to-engage-the-client-in-the-design-process-a73998ece46f

Turing, A. 1950. I.—Computing machinery and intelligence. Mind. p. 433-460.

Ubah, K. 2021. Learn Web Development Basics - HTML, CSS and JavaScript Explained for Beginners. Retrieved on 5 March 2023. Available at https://www.freecodecamp.org/news/html-css-and-javascript-explained-for-beginners/#:~:text=As%20a%20web%20de-veloper%2C%20the,just%20a%20design%20language%2C%20though.

Vaswani, A. & Shazeer, N. & Parmar, N. & Uszkoreit, J. & Jones, L. & Gomez, A. N. & Kaiser, L. & Polosukhin, I. 2017. Attention Is All You Need. Retrieved on 2 March 2023. Available at https://arxiv.org/abs/1706.03762

Vue. Introduction. Version 3 Retrieved on 6 March 2023. Available at https://vuejs.org/guide/introduction.html

Wickramasinghe, S. & Raza, M. DBMS: Database Management Systems Explained. In BMC Retrieved on 9 March 2023. Available at https://www.bmc.com/blogs/dbms-database-management-systems/

Wiesen, G. 2023. What is knowledge acquisition?. Retrieved on 27 February 2023. Available at https://www.languagehumanities.org/what-is-knowledge-acquisition.htm

Wiggers, K. 2021. 3 big problems with datasets in AI and machine learning. Retrieved on 1 March 2023. Available at https://venturebeat.com/uncategorized/3-big-problems-with-datasets-in-ai-and-machine-learning/

Wolff, R. 2021. Natural Language Processing (NLP): 7 Key Techniques. In MonkeyLearn. Retrieved on 10 March 2023. Available at https://monkeylearn.com/blog/natural-language-processing-techniques/

Wydmanski, W. 2022. What's the Difference Between Self-Attention and Attention in Transformer Architecture?. Retrieved on 2 March 2023. Available at https://medium.com/mlearning-ai/whats-the-difference-between-self-attention-and-attention-in-transformer-architecture-3780404382f3