



Design of a conversational recommender system in education

Stefano Valtolina¹ · Ricardo Anibal Matamoros² · Francesco Epifania²

Received: 26 June 2023 / Accepted in revised form: 1 April 2024
© The Author(s) 2024

Abstract

In recent years, we have seen a significant proliferation of e-learning platforms. E-learning platforms allow teachers to create digital courses in a more effective and time-saving way, but several flaws hinder their actual success. One main problem is that teachers have difficulties finding and combining open-access learning materials that match their specific needs precisely when there are so many to choose from. This paper proposes a new strategy for creating digital courses that use learning objects (LOs) as primary elements. The idea consists of using an intelligent chatbot to assist teachers in their activities. Defined using RASA technology, the chatbot asks for information about the course the teacher has to create based on her/his profile and needs. It suggests the best LOs and how to combine them according to their prerequisites and outcomes. A chatbot-based recommendation system provides suggestions through BERT, a machine-learning model based on Transformers, to define the semantic similarity between the entered data and the LOs metadata. In addition, the chatbot also suggests how to combine the LOs into a final learning path. Finally, the paper presents some preliminary results about tests carried out by teachers in creating their digital courses.

Keywords Conversational interface · Machine learning for education · Learning objects

✉ Stefano Valtolina
stefano.valtolina@unimi.it

Ricardo Anibal Matamoros
ricardo.matamoros@socialthingum.com

Francesco Epifania
francesco.epifania@socialthingum.com

¹ Department of Computer Science, Università degli Studi di Milano, Milan, Italy

² Social Things SRL, Milan, Italy

1 Introduction

Mainly due to the COVID-19 pandemic these last few years, we have witnessed an increase in the production of digital material in education. This exponential growth in educational content allows teachers to create courses accessible, even remotely, for their students. Moreover, thanks to developments in innovative fields such as artificial intelligence (AI), teachers can create and deliver digital material in a way we never imagined.

Learning management systems (LMS) are among the most used educational technologies based on artificial intelligence (AI). These systems can provide teachers with a new type of user intervention using recommendation mechanisms.

Several research works have demonstrated the efficacy of incorporating intelligent systems in the e-learning domain to address various tasks related to teaching and learning processes (Chen et al. 2020; Roll and Wylie 2016; Alam 2021). Specifically, these studies demonstrate how AI algorithms can support teachers in the educational process to personalise the courses according to the specific students' needs (Hwang et al. 2020). In particular, recommendation systems (RS) have emerged as one of the best solutions to address the information filtering problem to reduce the overload caused by information elaboration. These systems are the optimal tool to improve personalised learning by filtering didactic resources (Chen et al. 2020). As demonstrated (Valdez et al. 2016; Medio et al. 2020; Khanal et al. 2020), RS, such as collaborative filtering, content-based filtering, or hybrid, can be effectively applied to the e-learning domain.

Despite the importance of adopting intelligent LMSs, the main problem in using them is that many teachers are not well-trained to design proper digital courses. Some of them go for a “one size fits all” approach, which is rarely a recipe for success for what they expect students to know, understand or be able to do due to taking their course. Other teachers lack enough time to create digital courses, especially basic or professional ones. Finally, some LMSs lack effective assessment methods to gauge learning outcomes and the impact of the e-learning program both for teachers and students.

According to these considerations, the paper presents a new strategy to support teachers in creating a digital course for Italian schools and agencies using an e-learning platform named WhoTeach.¹ With digital courses, we mean an educational program delivered primarily through digital means, typically using online media and technologies. It consists of structured learning materials and resources accessible via the Internet. Learners can engage with the course content at their own pace and convenience, in our case, through an LMS.

The innovative idea consists of endowing the LMS with an intelligent chatbot to assist teachers in their activities by suggesting learning objects (LOs) as primary recommendation elements. Defined using RASA² (Bocklisch et al. 2017) technology, the chatbot offers the best LOs and how to combine them according to their prerequisites.

¹ <https://www.whoteach.it/> Last access: 2023-10-26.

² <https://rasa.com/> Last access: 2023-10-26.

In addition to suggesting how to connect the LOs, the chatbot explains why the module is significant. RASA is an open-source conversational AI framework that allows developers to build, deploy, and maintain chatbots and virtual assistants.

Finally, the paper presents some results of the tests carried out on the machine learning models used to predict the actions to be taken by the chatbot according to the semantic understanding of the teacher's messages and some preliminary results about tests carried out in creating their digital courses. In particular, we defined and implemented a model to evaluate teachers' level of acceptance and intention to use the chatbot, extending the UTAUT model (The Unified Theory of Acceptance and Use of Technology) presented in Venkatesh et al. (2003).

The paper is structured as follows. Section 2 describes the state of the art of using a conversation interface in education and explains how we used the chatbot to support teachers in creating a course. Section 3 describes the chatbot's recommendation system (RS) to suggest better LOs. Section 3 describes the recommendation system the chatbot uses to present better LOs. These LOs refer to self-contained educational content units retrieved by didactical repositories such as ARIADNE,³ NSDL⁴ and MERLOT.⁵

The Section explains how the chatbot indicates a sequence of proper LOs and the strategies for combining them. Section 4 presents some tests carried out in creating their digital courses. In particular, we defined and implemented a model to evaluate teachers' level of acceptance and intention to use the chatbot, extending the UTAUT model (The Unified Theory of Acceptance and Use of Technology) presented in Venkatesh et al. (2003). Finally, Sect. 5 sums up conclusions and future works.

2 Related work

2.1 Background about the use of chatbots in the education domain

Many studies regarding using chatbots in the educational domain exist in the literature. For example, authors in Okonkwo and Ade-Ibijola (2021); Medeiros et al. 2018; Smutny and Schreiberova 2020) present how to use conversational agents in areas such as teaching and learning, administrative assistance, assessment, consultancy and research and development.

According to the review (Okonkwo and Ade-Ibijola 2021), chatbots are mainly applied for teaching and learning (66%). They promote rapid access to materials by students and faculty at any time and place (Alias et al. 2019; Wu et al. 2020). This strategy helps save time and maximise students' learning abilities and results (Murad et al. 2019), stimulating and involving them more in teaching work (Lam et al. 2018). Furthermore, they can automate many student activities, including submitting homework, replying to emails and sending feedback on the courses followed (Deschênes 2020; Urdaneta-Ponte et al. 2021). Finally, the chatbot can be used as a real personal assistant for teachers, assisting them in their daily tasks.

³ <https://www.ariadnelearning.it/> Last access: 2023-10-26.

⁴ <https://nsdl.oercommons.org/> Last access: 2023-10-26.

⁵ <https://merlot.org/merlot/> Last access: 2023-10-26.

However, to our knowledge, chatbots are rarely used to assist teachers in creating new digital courses. Teachers can use authoring tools and learning platforms to develop and host online courses, such as Absorb,⁶ Learnopoly,⁷ Elucidat,⁸ Thinkific,⁹ Teachable,¹⁰ Podia,¹¹ or Learnworlds.¹² These tools help teachers create, launch and review an e-learning course, and the choice mainly depends on what teachers want about content to offer, hosting requirements, audience, and budget.

Nevertheless, these solutions cannot support teachers in creating digital courses, starting from learning material repositories available on the net or suggestions based on past colleagues' experiences to build on and create new solutions in new situations.

Our idea is to use a conversational agent that assumes the role of a prompter to assist teachers in finding and selecting proper learning open-access materials available on the Internet. One of the main benefits of integrating a chatbot into a Learning Management System (LMS) is that it can make educational processes easier for teachers. To validate our idea, we combined the chatbot in a learning platform, WhoTeach, to support teachers in creating a digital course for Italian schools and agencies. The platform uses learning objects (LOs) as building blocks to compose a course [17,19,209].

2.2 Recommendation systems in education

Reusing existing LOs is a valuable solution for helping teachers because they represent helpful material used in the past by other colleagues to create efficient courses.

Lately, the area of research dealing with finding and recommending a list of LOs that fit specific teachers' needs and requirements is very active (Wu et al. 2020; Urdaneta-Ponte et al. 2021). Nevertheless, teachers highlight difficulties in effectively combining small chunks of educational material to meet the teacher's academic needs (Murad et al. 2019; Campbell 2003). It is precisely for this reason that the conversation agent comes into play. To avoid the plethora of material available on the net becoming a disadvantage by paralysing the creation process, our chatbot-based recommendation system (RS) aims to find and suggest the right LOs, as explained in the next Section. In detail, we describe the motivation for using a chatbot to offer LOs and how learning resources can be appropriately assembled in a course meeting the teachers' objectives and requirements.

Several standards have emerged over the years to facilitate the sharing and reuse of learning materials, establishing metadata policies and providing suggestions for using the LOs. SCORM Sharable Content Object Reference Model (SCORM 2003) is a well-known example of a reference strategy for describing and cataloguing LOs. SCORM provides users with procedures for aggregating LOs and methods for processing contents on the related learning management system. To ensure interoperability,

⁶ <https://www.absorblms.com/> Last access: 2023-10-26.

⁷ <https://learnopoly.com/> Last access: 2023-10-26.

⁸ <https://www.elucidat.com/> Last access: 2023-10-26.

⁹ <https://www.thinkific.com/> Last access: 2023-10-26.

¹⁰ <https://teachable.com/> Last access: 2023-10-26.

¹¹ <https://www.podia.com/> Last access: 2023-10-26.

¹² <https://www.learnworlds.com/> Last access: 2023-10-26.

solutions like the one proposed by SCORM need to lever on sets of metadata that define standard guidelines for the classification of LOs. Examples of these metadata are Dublin Core¹³ and IEEE LOM.¹⁴

Research and surveys, such as (Dagienė et al. 2013; Hoebelheinrich et al. 2022), show how Dublin Core is suitable for describing the bibliographic side of digital resources, but LOM allows the best representation of the pedagogical aspects. Regardless of the metadata sets, these standards do not indicate which are more suitable for describing LOs according to given teachers' preferences. And even if some studies (Hoebelheinrich et al. 2022) recommend a minimal metadata set to describe learning material, they do not report specific rules to follow. Some attempts in this direction have led to adapting the metadata presented in the various standards into profiles that meet community context-specific needs (Palavitsinis et al. 2014). This approach has led to the emergence of the metadata Application Profile (AP) concept. An AP takes one or more standards as a starting point, imposing some restrictions and modifying the vocabularies, definitions or other elements of the original standard to adapt to the needs of a specific application (Duval et al. 2002).

For example, several Application Profiles implement the IEEE LOM and Dublin Core standards for describing learning resources ((Zschocke et al. 2009), UK LOM Core,¹⁵ and CanCore¹⁶), scientific resources (Darwin Core¹⁷), cultural resources (ESE¹⁸ and SWAP¹⁹) and more. The use of APs has allowed the birth of Learning Object Repositories (LOR) as examples of digital libraries or electronic databases where educators can share, use and reuse different LOs to build online learning opportunities for their students. Some of the best-known repositories regarding the number of titles collected are ARIADNE,²⁰ NSDL²¹ and MERLOT.²²

ARIADNE was founded in 1996 by the European Commission within the "Telematics for Education and Training" programme. The core of this infrastructure is a distributed library of digital and reusable educational components called the Knowledge Pool System (KPS). ARIADNE uses a set of metadata extrapolated from the General, Technical and Educational areas of the IEEE LOM. The user can search for resources through SILO (Search and Index Learning Objects), which allows simple, advanced or federated searches (through multiple repositories).

NSDL (The National Science, Mathematics, Engineering, and Technology Education Digital Library) was founded in 2000 by the National Science Foundation to provide a reference library for STEM-type resources. Users can create their profiles

¹³ <https://www.dublincore.org/> Last access: 2023-10-26.

¹⁴ <https://ieeexplore.ieee.org/document/9262118> Last access: 2023-10-26.

¹⁵ <http://www.ukoln.ac.uk/metadata/education/> Last access: 2023-10-26.

¹⁶ cancore.athabascau.ca/en Last access: 2023-10-26.

¹⁷ rs.tdwg.org/dwc Last access: 2023-10-26.

¹⁸ www.europeana.eu/ Last access: 2023-10-26.

¹⁹ www.ukoln.ac.uk/repositories/digirep/index/Scholarly_Works_Application_Profile Last access: 2023-10-26.

²⁰ <https://www.ariadnelearning.it/> Last access: 2023-10-26.

²¹ <https://nsdl.oercommons.org/> Last access: 2023-10-26.

²² <https://merlot.org/merlot/> Last access: 2023-10-26.

and get recommendations for LOs based on the user's previous interactions with the repository.

Finally, MERLOT (Multimedia Educational Resources for Learning and Online Teaching), born in 1997, is an open repository designed primarily for students and teachers. The peculiarity of the LOR is that it reviews the educational resources to help users understand whether the resource can function within their course. Reviews are composed of three dimensions: quality of the content, potential effectiveness as a teaching tool, and ease of use.

In training our chatbot-based recommendation system, we used these repositories and their related metadata as bases for implementing the retrieval strategy for creating a new digital course.

3 Design of the conversational agent's interaction strategy

In our work, we advocate using conversational agents not as student assistants but as mentors who can help teachers create digital courses.

The technology used to develop our chatbot is RASA (Bocklich et al. 2017), an open-source framework that creates text- and voice-based chatbots. Figure 1 presents the general architecture of the system. The main components are RASA NLU, i.e. the language comprehension engine, and RASA CORE, which predicts the following action to be performed following user input.

The RASA NLU component analyses the grammar and logic of the sentence the teacher enters, from which it extrapolates the parts that interest RASA CORE to elaborate the answer. RASA NLU is responsible for classifying the intents and extracting the entities.

Based on the context of the conversation, the RASA CORE chooses the following action to take or the response to give to the user. A dialogue management model needs to be trained to determine the next step. Through stories²³ and the rules, you can specify the training data of this model. The rules describe small parts of the conversation that must follow a specific path; on the contrary, stories allow you to create different directions of the conversation based on whether the user can respond in the way expected or can give unexpected answers.

If the following action to be performed is a "custom action", RASA CORE requests the action from the "RASA Action Server", which executes it and returns the resulting response or event. Once the server retrieves the data relating to the course to create, these are integrated into a JSON object within a "custom action" and sent to the Recommender System (RS)Server. The RS server returns the list of LOs extracted from the dataset according to the received information.

The Tracker store is the database where the conversations of the virtual assistant are saved. In contrast, the Lock Store serves to process the messages in the correct order and to lock the conversations. The messages are actively processed, allowing the operation of multiple RASA SERVERS simultaneously.

²³ Rasa stories. <https://rasa.com/docs/rasa/stories> Last access: 2023-10-26.



Fig. 1 General architecture of the e-learning platform

The RS Server adopts a machine learning model called BERT (Devlin et al. 2018), which uses Transformers as the deep learning model, introduced in 2017 by Google, to realise various Natural Language Processing tasks, including translation, classification and text synthesis. Among the functions that BERT can do, one of the most important is Semantic Text Similarity (STS), which calculates the semantic similarity between two input texts. Even if BERT achieves high accuracy in this task, two complete texts (not only metadata, as happens in our case) must be entered into the model, resulting in a substantial computational overhead. To solve this problem, we adopted Sentence-BERT-S-BERT (Reimers and Gurevych 2019), a modification of BERT that uses Siamese neural networks to generate sentence embeddings that capture the meaning of the two sentences. Subsequently, the generated embeddings are compared through Cosine Similarity to obtain the semantic similarity between the two sentences considered.

In particular, in the context of our work, the chatbot-based recommendation system uses S-BERT to identify the LOs with the most semantically similar metadata to the

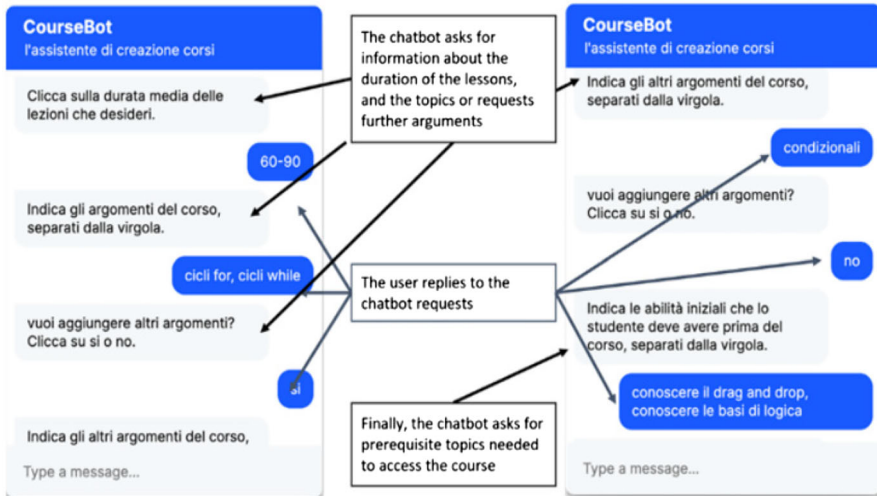


Fig. 2 Two screenshots present the chatbot interaction. In the first one, the chatbot asks the teacher for information about creating a programming course. In detail, it asks to insert: 1. The average time for each lesson, 2. The number of topics to cover, and 3. If the teacher wants to add new topics. In this example, the teacher responds “yes” and then inserts a new topic: “conditional structures”. On the right, in the final request, the chatbot asks to insert prerequisites the students need to know before taking the course. The teacher indicates that the students must know how to drag and drop and the basics of first-order logic

data entered by the teacher via chatbot. Data that the teacher specifies for describing the course to create.

3.1 The whoteach learning platform

Figures 2, 3, 4 present screenshots of the learning platform, WhoTeach, we used to integrate the chatbot. Figure 2 depicts the situation in which the teacher interacts with the chatbot to specify information about the course to create, such as its difficulty and the duration of the lessons.

Regarding topics, skills and competencies, the chatbot allows adding items to the corresponding list if the user needs it. Once the teacher has entered the course information, the parsing system checks the orthographic and performs the translation into English (since the dataset is in English) via the DeepTranslator library.²⁴ Subsequently, S-BERT transforms the topics, difficulty, type and duration fields into embeddings and tensors. Later, the similarity between the tensors of this information and the tensors of each LO’s metadata is measured using cosine similarity. The function used to compute this metric is the *semantic_search* from the Sentence-Transformers library.²⁵

All the scores obtained are entered into a data frame. Finally, to calculate the final SCORE, the system uses the average of the scores of the individual metadata for each

²⁴ Deep translator. <https://pypi.org/project/deep-translator/> Last access: 2023-10-26.

²⁵ Sentence-transformers. <https://www.sbert.net> Last access: 2023-10-26.



Fig. 3 In the screenshot on the left, the chatbot asks the teacher to select the LOs to insert into the final course. Clicking on a LO on the left allows it to see further information about it: the topics the LO covers, a description, the rating and its keywords (including accessibility indications)

LO. The chatbot recommends the LOs to the user based on their SCORE and presented via a graphical interface.

Figure 3 shows the checkboxes for the teacher to select the desired LOs and the resources she/he wants to include in the digital course. If a LO contains exercises, the teacher can specify to repeat the activity if the student fails to finish a LO in the desired time or according to an established rating.

Nevertheless, finding suitable LOs is insufficient because a course is not an arbitrary sequence of LOs but a combination of them based on pedagogical relationships. A prerequisite relationship exists between two LOs if one is significantly helpful in understanding the other. Therefore, once the teacher selects a set of LOs, the chatbot suggests how to combine them in the proper sequence and provides other related LOs that the teacher can use to define a final course.

These suggestions concern possible LOs to integrate into the final course. In Fig. 4, you can see the suggested LOs in the first line at the top of the dashboard. The white dashed rectangles represent the suggested LOs the teacher has not selected but that the system provides because they relate to the other LOs. The teacher can select them in a second moment. In the Figure, the teacher has chosen LOs that are coloured, and the colours allow her/him to discriminate the type of learning content (video lesson, exercise, document and quiz). Finally, the teacher can sequence the LOs by dragging and dropping them on the third line of the screenshot.

Another important suggestion is present in the second line of the screenshot in Fig. 4. In this case, the recommendation system suggests how to combine the LOs in different learning paths. The branches depend on the output of the previous LO. If the LO ends with a test, the choice of the next LO depends on the obtained result. Without a test, the choice depends on the student's evaluation.

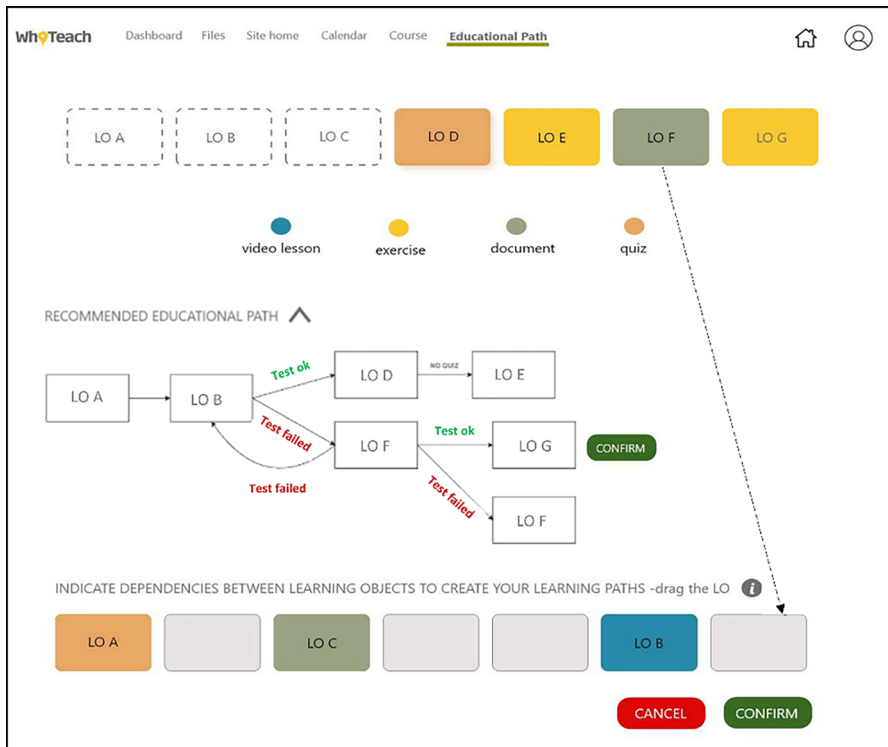


Fig. 4 The system presents a list of LOs selected by the teacher at the top of the screenshot. The teacher does not specify the white dashed LOs. The teacher can sequence the LOs by dragging and dropping them on the third line of the screenshot. Finally, as depicted in the second line of the screenshot, the RS can suggest a possible combination of LOs presented in the first list showing different learning paths. As explained in the text, the branches depend on the output of the previous LO

3.2 Explanatory scenario

Mrs. Smith, a middle school science teacher, wants to create a new "Basic Astronomy" course. She uses the chatbot-based recommendation system to find suitable Learning Objects (LOs) for her course.

- *Purpose of the System:* Mrs Smith interacts with the chatbot, asking for LOs related to "Basic Astronomy".
- *Technical Details:* The chatbot uses the S-BERT model to understand Mrs Smith's request and match it with relevant LOs. For instance, Mrs. Smith's request "Tell me about stars" and an LO titled "Introduction to Stars" might be transformed into similar embedding vectors, indicating a match.
- *Metadata specification:* Mrs Smith specifies she wants LOs with a duration of 10-15 min for students aged 12-14, with difficulty: beginner, language: English, type: video and with the related keywords: Stars, Planets, Moon.

- *Conversational Assistant*: The chatbot, implemented with RASA, understands Mrs Smith's specifications and maps them to the metadata profile.
- *Filtering Steps*: First Step: The system filters out LOs based on the language (English). Let's say out of 100 LOs, 70 are in English. Second Step: From these 70 LOs, the system further filters them according to difficulty, duration, keywords, and age. Let's assume 20 LOs match all of Mrs Smith's criteria. Third Step: The system then ranks these 20 LOs based on how closely they match Mrs Smith's preferred language metadata. The top 5 LOs are selected.
- *Recommendation System's Next Step*: The system identifies that the "Introduction to Stars" LO has a prerequisite LO titled "Basics of the Universe". It then suggests that Mrs Smith use "Basics of the Universe" before "Introduction to Stars" to create a coherent learning path for her students.

Mrs Smith has a set of 5 LOs, presented logically to help her make her "Basic Astronomy" course.

3.3 Implementation and deployment

Initially, we used a traditional approach to associate the value of entities extracted from the user message using the RASA NLU model. Since the different intents are made up of data that are very similar to each other or even the same (for example, numerical data such as the age of the students, the level of difficulty of the lessons, the number of lessons, etc.), using this solution the NLU model confused them.

To solve this problem, we have chosen to map user responses to textual intents. We used the RASA validators to ensure that the extraction model associated only the significant parts of the message with the string. Thanks to these custom actions, extracting the significant components from the strings entered by the user and preventing an incorrect value from being associated with the message is possible.

To make the virtual assistant available to the public, we used Docker²⁶ and Kubernetes,²⁷ as suggested in the RASA documentation.²⁸ From a practical point of view, we created the following Docker images for the architectural components: Chatbot, RASA Action Server and Recommender System Server. We then used the tools made available by Google Cloud Platform to make the web page containing the chatbot accessible via IP address. We saved the Docker images in the Artifact Registry,²⁹ the Google repository service that stores and organises Docker images. Subsequently, within the cluster management system provided by Google, Google Kubernetes Engine(GKE),³⁰ we created a Kubernetes cluster, and the three PODs were placed inside it, which contain the containers associated with the three images made (each POD contains only one container as described in Fig. 4).

We defined a set of Load Balancer Services for the chatbot and the Recommender System Server to allow other services to access them outside the cluster. In Fig. 5,

²⁶ Docker. <https://www.docker.com/> Last access: 2023-10-26.

²⁷ Kubernetes. <https://kubernetes.io/> Last access: 2023-10-26.

²⁸ Rasa deployment. <https://rasa.com/docs/rasa/deploy/introduction> Last access: 2023-10-26.

²⁹ Artifact Registry. <https://cloud.google.com/artifact-registry> Last access: 2023-10-26.

³⁰ Google Kubernetes Engine. <https://cloud.google.com/kubernetes-engine> Last access: 2023-10-26.

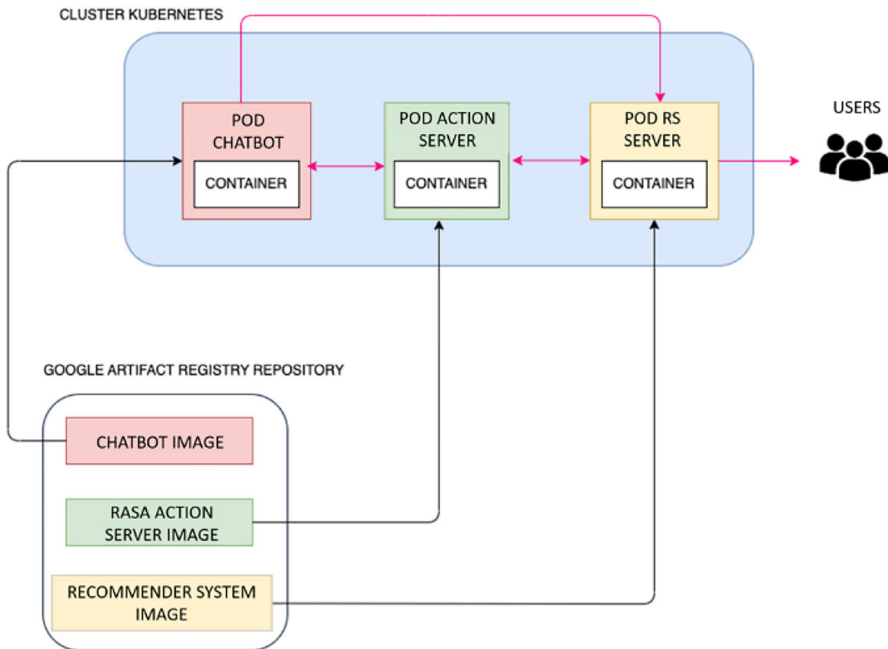


Fig. 5 Cloud architecture of the e-learning platform

which shows the cloud architecture, the chatbot POD is connected to the RS Server POD as the former's IP address is entered in the chat widget script contained within the HTML page, accessed via a request to the Recommender System Server. Instead, to allow the internal connection between the chatbot and the RASA Action Server and between the RASA Action Server and the Recommender System Server, two Services have been created with InternalTrafficPolicy of the local type, which allows you to use only local endpoints for internal traffic to the cluster.

4 Conversational recommender system at work

4.1 How the conversational agent suggests LOs

Providing teachers with a set of LOs helpful in creating a new course is the responsibility of the chatbot-based recommendation system, whose interaction strategy we presented in Sect. 3.

From a technical point of view, as said before, the recommendation service at the base of the conversational agent uses a generation of embedding representations through the application of the S-BERT model. This model works on two input sentences, a and b , and through a B function, transforms them into two vectors embedding representations \vec{a} and \vec{b} , helpful for calculating the similarity score y_{sim} . S-BERT

uses the pre-trained BERT model as the B function for the actual generation of embedding and applies a Mean Pooling layer on each B output to calculate the y_{sim} value in the output. In our context, the set of metadata M of the learning objects we used are:

- Duration
- Age
- Difficulty
- Language
- Type
- Keywords

The part of the conversational assistant implemented with RASA allows mapping the teacher's specifications on a profile based on the list of metadata M, representing the user's preferences.

We use this set to apply three filtering steps on the set of LOs. The first filtering step is defined as:

$$L^1 = \{LO \in L | \text{similarity}(B(m_{\text{language}_i}), B(m)) > k, \forall m \in M_{\text{language}}\} \quad (1)$$

where $L^1 \subset L$ is obtained by calculating the similarity between the values m_{language} entered by the teacher and each value $B(m)$ of the language metadata, such that $m \in M_{\text{language}}$. The second step applies the B function on the metadata $m_{\text{difficulty}}$, m_{duration} , m_{keywords} , and m_{age} to use S – BERT on the set L^1 . We defined this process as:

$$L^2 = \{LO \in L^1 | F((f_{\text{avg}}(B(m_i)), f_{\text{avg}}(B(m)) | \Theta) > c, \forall m \in M_{\text{language}})\} \quad (2)$$

where k and c are similarity thresholds fixed a priori. In particular, S-Bert applies an average pooling function f_{avg} to the output of B , on both input values m_i , m . The function F corresponds to a multi-layer perceptron where Θ represents the set of learnable parameters of the model, and through a σ activation function used as the output layer, it returns the similarity score between the two input values m_i , m .

The third step consists of applying formula (1) to the metadata values m_{language} listed as preferred by the teacher to obtain the set of LO L^3 . The top_5 LOs from this set are returned to the teacher and displayed via the conversational agent interface.

As explained in the next Section, once we have identified the LOs that meet the teacher's needs, the recommendation system has to leverage the prerequisites that link each LO to the others to suggest how to create the final learning path.

4.2 Model for prerequisite extraction

Discovering the pedagogical relationships between LOs is a complex and time-consuming practice, usually performed by domain experts. A prerequisite relationship is a pedagogical relationship that indicates the order in which we can propose the concepts to the user. In other words, we can say that a prerequisite relationship exists between two concepts if one is significantly helpful in understanding the other.

Our recommendation system computes a list of LOs to be recommended, which it sends to the prerequisite analysis model. The model returns the ordering of the LOs according to the information concepts contained.

For this prerequisite extraction model, we used an innovative approach based on deep learning to automatically identify the prerequisite relationships between concepts to create pedagogically motivated LO sequences. The model exclusively exploits linguistic characteristics extracted from the LO description that each LO repository provides. Considering only textual content is perhaps the most complex condition for inferring relationships between educational concepts since it cannot rely on structured information. At the same time, this is also the closest condition to a real-world scenario.

The protocol we implemented aims at the first stage to extract five topics that better represent the LO. We can infer the topics using the LO description. In the second phase, we use the five topics to find the five corresponding Wikipedia pages in which each topic is explained.

Then, we have to choose the wiki page that exhibits the highest similarity to the original description of the LO. This activity allows us to characterise the LO content better for inferring the pedagogical relationships that link it to the others. However, a lone wiki page linked to a Learning Objective (LO) may not comprehensively grasp the content of the LO.

For this reason, in phase three, we need to investigate if other wiki pages can better describe the LO content. To do it, we calculate the similarity of the LO description with all the Wikipedia pages in our dataset. Our dataset comprises the wiki pages linked to all topics associated with all LOs we consider in our prerequisite extraction model. Once we have finalised this step, we must choose which wiki page better describes the LO content. The wiki page we identified at stage two or three. We select the wiki page with the minor similarity score concerning the LO description to determine this final mapping using a cosine similarity distance.

Once each LO is linked to the best wiki page to describe its content in detail, we need to define when an LO is a prerequisite of another.

In this fourth phase, our model aims to learn if a LO "A" is a prerequisite for a LO "B" by analysing the related wiki pages. The proposed model implements the identification of prerequisite relationships between concepts in the Italian language and exploits the approach proposed in Angel et al. (2020) for the PRELEARN task of the EVALITA 2020 project. Following, we report a pseudo-code description of the algorithm we used to implement our prerequisite extraction model.

4.2.1 Phase 1—topic extraction

The first step aims at extracting the topics most representative of each LO. This task is based on a Topic Modelling approach. Topic Modelling is an unsupervised ML method that receives in input a corpus of documents and extracts the most relevant topics and concepts to describe the vast volumes of data in a reduced dimension, thanks to the identification of hidden concepts, relevant characteristics, latent variables and semantic structure of the data.

In particular, we used the Latent Dirichlet Allocation proposed by David Blei, Andrew Ng and Micheal I. Jordan in Blei et al. (2001) as a topic modelling approach to discover the underlying topics to link to each LO in a collection of text documents. The main idea of the model is the assumption that each document is a mixture of a fixed number of topics, and each topic is a probability distribution of words. The algorithm applies an inference process to determine the topic distribution in the documents (in our case, the LO descriptions) and the word distribution in each topic. We denote V as the vocabulary size, N as the number of words in a document, and M as the number of documents in the corpus D . Then, the model assumes the following process for each document $w \in D$.

- Choose $N \sim \text{Poisson}(\xi)$
- Choose $\theta \sim \text{Dir}(\alpha)$
- For each word w_n
 - Choose a topic $z_n \sim \text{Multinomial}(\theta)$
 - Choose a word w_n from $p(w_n | z_n, \beta)$, a multinomial probability conditioned on the topic z_n

Then, fixing hyperparameters α and β , we can extract the posterior probability of the corpus topics with the following formula based on a Bayesian process.

$$p(D, \alpha, \beta) = \prod_{d=1}^M \int p(\theta_d | \alpha) \left(\prod_{n=1}^{N_d} \sum_{z_{dn}} p(z_{dn} | \theta_d) p(w_{dn} | z_{dn}, \beta) \right) d\theta_d \quad (3)$$

After completing this step, we can associate each LOt with the five main topics that represent it.

(phase 1 pseudo-code from a single document)

```
FUNCTION ExtractTopics(document: String) -> List[topics]:
  N = Poisson(ξ) # Choose N ~ Poisson(ξ)
  θ = Dirichlet(α)
  for each word in document:
    zn = Multinomial(θ)
      # Choose a topic zn ~ Multinomial(θ)
    wn = Multinomial(p(wn | zn, β))
      # Choose a word wn from p(wn | zn, β)
  # Posterior probability calculation
  posterior_probability = calculate_posterior_probability(zn, wn, Alpha, Beta)
  RETURN posterior_probability
```

4.2.2 Phase 2—Wikipedia page extraction

After we have found the five keywords that best represent the most relevant topics for each LO, we exploit them to search for corresponding five Wikipedia pages that

can be used to explain them. To accomplish this, we employ the Python library called "Wikipedia-API",³¹ which allows us to extract various information from Wikipedia. The API will enable us to identify the better five wiki pages to associate with an LO that we can use to characterise the LO content.

Once we have linked each LO to five distinct Wikipedia pages, we need to establish the page that exhibits the highest similarity to the original description of the LO. To do it, we chose to employ the cosine similarity metric. This metric is widely used in Natural Language Processing to gauge the similarity between two vectors based on their angle in a high-dimensional space. Therefore, adopting this approach, we can calculate the distance between the LO description and the associated Wikipedia pages to select the page with the minimum distance as the most similar.

(phase 2 pseudo-code)

```
FUNCTION SearchWikipediaPages (topics: List[topics]) ->
List[wikiPages]:
    wikiPages = []
    FOR topic in topics:
        page = WikipediaAPI.search(topic)
        wikiPages.append(page)
    RETURN wikiPages
```

4.2.3 Phase 3—similarity evaluation of LO with all the extracted wiki pages

We cannot consider only a single wiki page as the best candidate to describe LO content. We need to find if, in our dataset, we can use other wiki pages for our purpose. We created this dataset by inserting all wiki pages related to all topics of all LOs of our repository. This approach allows us to narrow our focus to a subset of Wikipedia pages relevant to our domain. Using our dataset, we can calculate the cosine distance between the LO description and all the Wikipedia pages. This step aims to identify any additional relevant pages associated with the LO that may have been overlooked previously.

To accomplish this, we must locate the nearest neighbours for each LO description in a high-dimensional space. Instead of using a conventional algorithm like K-Nearest Neighbours for this task, we adopted an approach based on Approximate Nearest Neighbours (Indyk and Motwani 1998) to address this challenge. This family of methods presents a solution to address the challenge of high dimensionality by intelligently partitioning the vector space. As a result, we can limit our analysis to a smaller subset of the original set, easing the computational burden.

Once we have finalised this step, we have two options for the best Wikipedia page to associate with a LO: the wiki page identified at phase 2 or found at this stage. To determine the final mapping, we select the option with the closest cosine distance between the LO description and the respective Wikipedia page.

³¹ <https://pypi.org/project/Wikipedia-API/>. Last access: 2023-10-26.

(phase 3 pseudo-code)

```

FUNCTION FindBestMatch(description: String, wikiPages:
List[wikiPages]) -> bestWikiPage:
  bestScore = INFINITY
  bestWikiPage = NULL
  FOR page in wikiPages:
    score = CosineSimilarity(description, page.content)
    IF score < bestScore:
      bestScore = score
      bestWikiPage = page
  RETURN bestWikiPage

```

4.2.4 Phase 4—prerequisite extraction based on EVALITA 2020 and BERT

Our proposed model draws inspiration from the work of Angel et al. in the PRELEARN (Prerequisite Relation Learning) domain shared task of EVALITA 2020 (Angel et al. 2020). EVALITA is a periodic campaign to advance language and speech technology for the Italian language, and their model demonstrated the best performance. Specifically, the PRELEARN task focused on classifying whether a pair of concepts exhibited a prerequisite relationship. The dataset analysed in this context was ITA-PREREQ, which contains pairs of Italian concepts connected by a prerequisite relation. In particular, each row of the dataset presents:

- Wikipedia page associated with concept A
- Wikipedia page associated with concept B
- Label: 1 if B is a prerequisite of A, 0 otherwise

Among the proposed solutions, the most effective one involved encoding all the concept pairs using an Italian model called BERT2 (Devlin et al. 2019), which we have fine-tuned on the training dataset of wiki pages mentioned earlier. Subsequently, we used a single-layer neural network to map the 1536 features (768 for each of the two vectors generated for the wiki pages) produced by the BERT algorithm in the encoding space to an output space with a dimension of 2. This mapping represents the two possible classes: the existence of a prerequisite relation or the non-existence of such a relation.

Similarly to the process just described, our approach comprises two steps. First, we fine-tuned BERT on the ITA-PREREQ dataset to obtain the 768-dimensional vector that represents the combination of the concept associated with each its Wikipedia description and related LO. Since we started with a dataset that contains LOs not labelled according to the prerequisite relationships, we cannot train the BERT model in a supervised way. Therefore, we used the standard pre-trained BERT model to fine-tune the EVALITA dataset. Subsequently, the model we obtained from this fine-tuning is used to make inferences on our dataset. The aim was to infer whether A is a prerequisite of B or not for a pair of LOs. Specifically, we applied the pre-trained encoding model alongside a single-layer dense neural network to our LO dataset. Given a pair of LO, the model receives the two Wikipedia pages associated with them,

determined in the previous steps, then concatenates their embeddings and applies the dense neural network to output the binary label that represents the existence of a prerequisite relation between LOs.

(phase 4 pseudo-code)

```
FUNCTION CheckPrerequisite(wikiPage_A: WikiPage, wikiPage_B: WikiPage) -> Boolean:
    vector_A = BERT_Encode(wikiPage_A.content)
    vector_B = BERT_Encode(wikiPage_B.content)
    relation = NeuralNetwork.predict(vector_A, vector_B)
    RETURN relation == PREREQUISITE
```

5 Validation of the chatbot assistant

5.1 Evaluation of the RASA models

To measure the ability of the virtual assistant to understand what the user writes while interacting with the chatbot, we tested the NLU model through a tenfold cross-validation. The resulting accuracy is 87%. The confusion matrix in Fig. 6 reports how often the intents were confused with other intents by RASA NLU. As seen in the matrix, the intents relating to skills and abilities, having similar example sentences, tend to be confused; the same goes for the course topics and names.

The numeric fields (age, number of lessons, formats, video lessons, exercises, quizzes, documents), being very similar to each other, if considered separately, would lead to a performance unsatisfactory of the model, which would not reflect the accurate good functioning of the chatbot. For this very reason, we decided to group them in a single intent to eliminate the risk of confusion and match the result of the evaluation of the NLU model with the actual efficiency of the chatbot.

Moreover, the confusion matrix in Fig. 6 gives us an insight into the excellent performance of our NLU model in identifying intents. Notably, the intent "inform numerics" in Fig. 7 was identified with high accuracy, with 78 correct predictions without any deviation. This result indicates the robustness and effectiveness of our model in this domain. Sometimes, the intent "inform ability" was confused with "inform skills". However, the model presents a reasonable degree of discriminability to discriminate between subtle semantic nuances. The intent "inform topics" shows good accuracy, with minimal confusion with "inform course name", which we can quickly address with further optimisation. Overall, the confusion matrix reflects the high quality of our NLU model and its ability to classify intents with reasonable accuracy, making it a reliable tool for conversational agents in the educational domain.

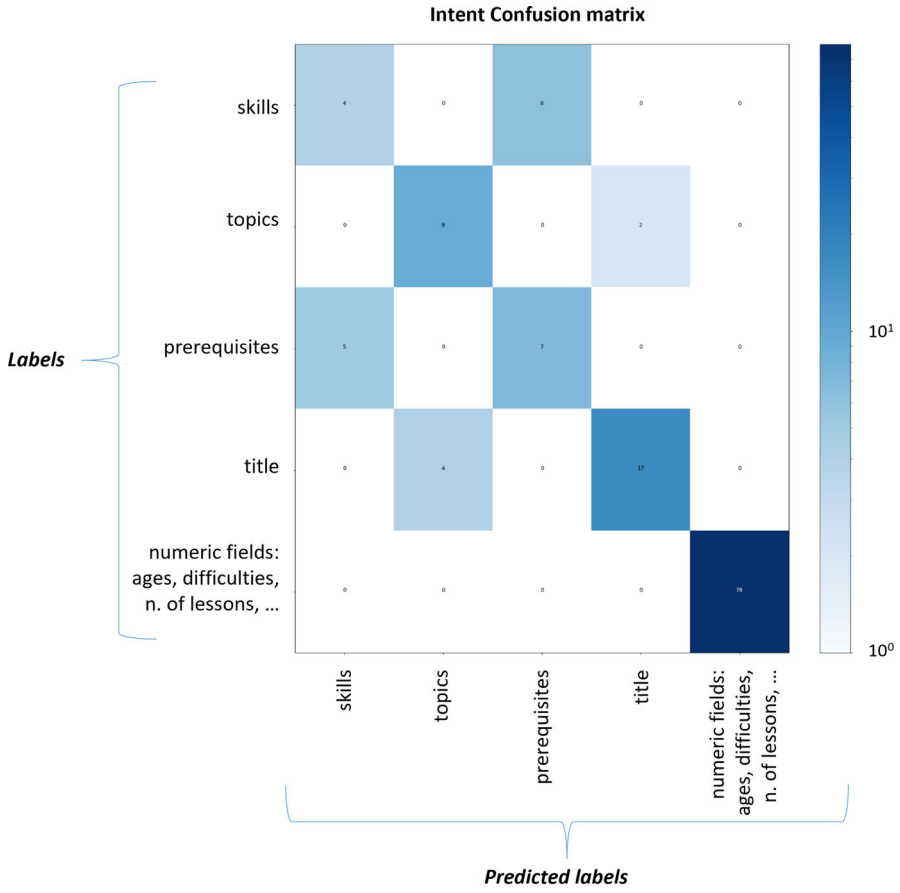


Fig. 6 Intent confusion matrix of the NLU model

| True Label | Predicted Label | | | | |
|-----------------|-----------------|--------|---------------|-------|-----------------|
| | Skills | topics | Prerequisites | title | inform numerics |
| Skills | 4 | 0 | 6 | 0 | 0 |
| topics | 0 | 9 | 0 | 2 | 0 |
| Prerequisites | 5 | 0 | 7 | 0 | 0 |
| title | 0 | 4 | 0 | 17 | 0 |
| inform numerics | 0 | 0 | 0 | 0 | 78 |

Fig. 7 Table used to design the Intent confusion matrix in Fig. 6 ("inform numerics" is used to identify the numeric fields in the confusion matrix)



Fig. 8 Charts of critical issues for usability tests for the first (on the top) and second (on the bottom) analysis. The tasks are: Task 1: Log in to the platform, Task 2: Create a new course, Task 3: Add a LO, Task 4: Make a LO invisible, Task 5: Edit the details of a LO, Task 6: Rename a LO, Task 7: Adding course completion criteria, Task 8: Changing the course end date, Task 9: Creating a group of students for a course, Task 10: Entering a password to access a course, Task 11: View the number of students who have accessed a resource, Task 12: View the courses a teacher teaches

5.2 Evaluation of the dashboard and the conversational recommendation system

In 2022, we carried out a study to improve the usability of the Learning Management System (LMS) component of the WhoTeach dashboard to allow users to use it faster and more intuitively without needing assistance.

We have used Benyon's 12 heuristics (Preece et al. 1994) to evaluate the platform: visibility, coherence, familiarity, clarity or affordance, navigation, control, feedback, restoration, constraints, flexibility, style and conviviality.

We carried out the test involving four usability experts, former students of the master's degree in communication theory and technology of the University of Milano, who tested the platform by impersonating a teacher.

After signing an informed consent document, the evaluators with proper accounts used the dashboard to investigate critical issues, the location where they were found,

their severity and which heuristics were violated. Severity is based on a scale of 1 to 3, where 1 is mild, and 3 is blocking. The tasks consist of 12 simple activities related to creating a course.

This analysis led to the discovery of numerous usability problems, for which we proposed several prototypes to fix them. After the society "Social Things," the owner of WhoTeach, approved the prototypes, we proceeded to implement them into the platform. Finally, we tested the platform again with the same initial methodology, and the result obtained was a clear improvement in the platform's usability.

The following Figure 8 shows for each task how many evaluators have found serious critical issues (red), mild critical issues (yellow) or no critical issues (green) during the first and the second tests. Only for tasks 7, 10 and 11 we can see that we still have critical issues, but although the result for task 11 remained unchanged, tasks 7 and 10 now present a lower percentage of evaluators who had critical issues and in both critical issues, they are only slight.

At the end of the tests, we administered the SUS questionnaires to the usability test participants. The arithmetic average obtained from the new SUS scores is 78, higher than the SUS average, which is 68 and is significantly higher than the result obtained during the first phase of 50.1. After adding the new score to the Sauro graph (Will 2017), as in Fig. 9, we can say that evaluators are pretty satisfied with the platform. The new platform is considered acceptable, and the associated adjective is good. Furthermore, compared to the previous grade, which resulted in an F, it obtained a B after the changes.

In 2023, we used the Conversational Recommenders System based on our chatbot to design an IFTS course (Istruzione e Formazione Tecnica Superiore—Higher Technical Education and Training). IFTS courses are training courses lasting one year that offer valuable and concrete tools to respond to the demands of the world of work as they align with the professional needs of companies. The training programme focused on the dynamics of the "Twin Revolution", using Python exercises to illustrate how data management can lead an organisation towards such a revolution.

We entrusted the initial course design to an expert course designer, who we invited to structure the course based on his expertise and resources. He was then allowed to use

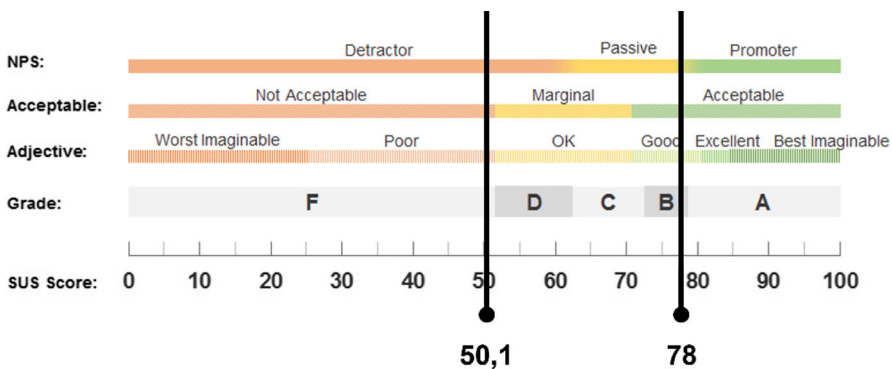


Fig. 9 Average SUS score of the two tests on Jeff Sauroe's graph

the Conversational Recommender System: using this system, the instructor outlined the course structure and integrated the first four Learning Objects related to the Twin Revolution, focusing mainly on theoretical aspects.

Specifically, the first four LOs used are PowerPoint presentations with the following titles: "Introduction to the Data-Driven Economy and the Twin Revolution", "The Data-Driven Business and the Main Technical Roles in the Context of the Twin Revolution", "Data-Driven and Application Cases in the Context of the Ecological Transition" and "Technologies for the Data-Driven Approach". Later, through a second interaction with the system, he enriched the course with LOs offering Python exercises related to IoT data management.

Regarding efficiency, while the instructor's independent preparation of a 12-h course took 32 h, using the conversational recommender system reduced this time to 4 h, followed by another 4 h for quality verification of the selected LOs. When asked for student feedback on the course material at the end of the course, the teacher reported that 18 out of 20 students found the material to meet their educational expectations. The remaining 2 suggested possible improvements.

5.3 Acceptance and intention to use the chatbot

To evaluate teachers' level of acceptance and intention to use the chatbot, we used the UTAUT model (The Unified Theory of Acceptance and Use of Technology) (Brocklisch et al. 2017), as also described in Valtolina and Matamoros (2023). This model includes eight user acceptance indicators that are well-validated in the context of many studies. The UTAUT model presents four significant constructs as direct determinants of user acceptance and intent to use a new technology: 1. Performance Expectancy (PE); 2. Effort Expectancy (EE); 3. Social Influences (SI); 4. Facilitating Conditions (FC).

Performance Expectancy measures how much an individual considers a valuable system for improving their job performance. Effort Expectancy measures how easy a system is to use. Social influence measures the influence of colleagues, instructors, and friends on the intention to use a new technology (Venkatesh and Davis 2000; Warshaw 1980). Finally, Facilitating Conditions measure how much external aid can facilitate the adoption and use of the system.

UTAUT is a generic acceptance analysis model which can be applied to different fields. To obtain a higher level of detail and a specific adaptation to our context, according to the results of these works (Venkatesh and Davis 2000; Venkatesh et al. 2012), we decided to use an extension model that integrates the primary constructs of the standard UTAUT model by adding these constructs: 1. Hedonic Motivation (HM); 2. Habit (H); 3. Trust (T).

The first construct measures the degree of appreciation of the system by users and how this could affect the intention to use it in the future. The second measures how much experience and the habit of using new technology can be helpful in its more concrete acceptance (Venkatesh et al. 2012). Finally, the Trust construct measures how much trust in the chatbot can affect its acceptance and future use.

Each construct can be related to others, specifying the hypothesis we need to study. As depicted in Fig. 10, each construct is related to other constructs defining the hypoth-

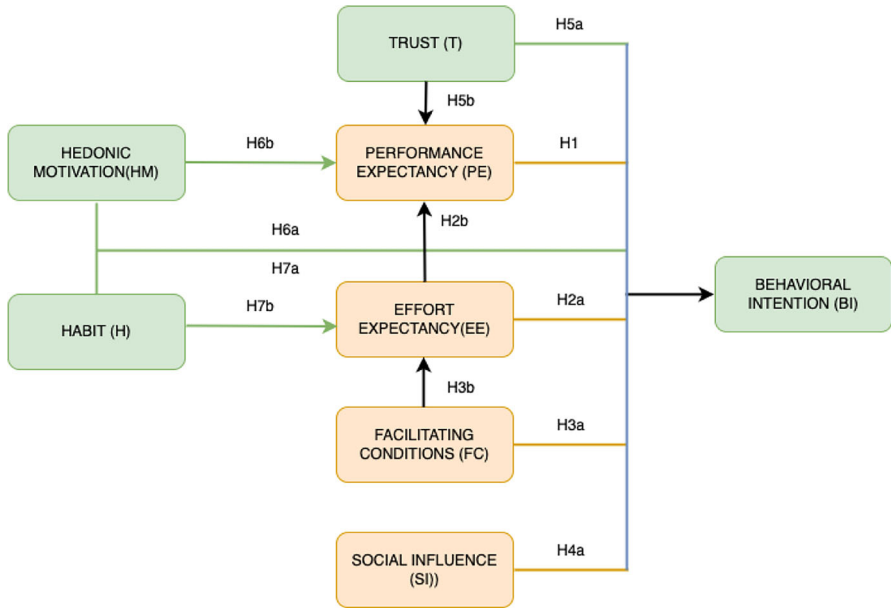


Fig. 10 Hypotheses schema. Each arrow represents a hypothesis that measures how much a construct can affect the validation of the other. For example, H6b, linking HM to PE, measures how much the degree of the chatbot appreciation can influence how much teachers consider it valuable for improving their job performance

esis we need to check. For example, Hypothesis 1 (H1) links PE to Behavioural Intention (BI) for evaluating how much the performance expectancy positively affects teachers’ intention to use the suggestions of the digital assistant. Or again, Hypothesis 6b (H6b), linking HM to PE, measures how much the degree of the chatbot appreciation can influence how much teachers consider it valuable for improving their job performance. We present all hypotheses in Fig. 10.

Twenty-six people participated in the evaluation, mainly chosen among SocialThingum³² company employees (twelve teachers of training courses in companies) and Computer Science department students of the University of Milan. The idea was to ask testers to create introductory programming courses in Italian. After administering a cognitive questionnaire, we can say that the participants were aged between 23 and 26 and had a master’s or three-year degree in Computer Science. Most of them stated that they occasionally use chatbots and that many have had the opportunity to take advantage of e-learning platforms such as Moodle.³³ Finally, the participants demonstrated solid knowledge of basic programming. During the test, we asked participants to create an introductory course for basic programming. At the end of the test, we provided testers with a questionnaire related to the constructs of our UTAUT model. With a total of 24 questions (5 for measuring PE, 3 for EE, SI, FC, T, and HM, 2 for H, and BI), each question tries to investigate how much the user

³² <https://www.socialthingum.it/> Last access: 2023-10-26.

³³ <https://moodle.org/> Last access: 2023-10-26.

| | Average | lia | SD |
|-----|---------|-----|-------|
| PE1 | 3.73 | | 1.185 |
| PE2 | 3.85 | | 1.120 |
| PE3 | 3.77 | | 1.243 |
| PE4 | 3.73 | | 1.041 |
| PE5 | 3.76 | | 1.165 |
| EE1 | 3.96 | | 1.038 |
| EE2 | 4.15 | | 0.925 |
| EE3 | 4.31 | | 0.970 |
| SI1 | 3.62 | | 0.804 |
| SI2 | 3.85 | | 0.881 |
| SI3 | 3.58 | | 1.102 |
| FC1 | 4.27 | | 0.962 |
| FC2 | 4.36 | | 0.757 |
| FC3 | 3.42 | | 1.238 |
| T1 | 3.27 | | 1.218 |
| T2 | 3.62 | | 1.061 |
| T3 | 3.62 | | 0.983 |
| HM1 | 3.58 | | 1.137 |
| HM2 | 2.88 | | 1.143 |
| HM3 | 3.77 | | 0.992 |
| H1 | 2.19 | | 1.021 |
| H2 | 1.62 | | 1.061 |
| BI1 | 2.81 | | 1.386 |
| BI2 | 2.85 | | 1.347 |

A

| Hypothesis | β | p-value | Acceptance |
|--------------------------|---------|---------|-----------------|
| H1: PE \rightarrow BI | 0.218 | 0.014 | Accepted |
| H2a: EE \rightarrow BI | 0.238 | 0.170 | Not significant |
| H2b: EE \rightarrow PE | 0.847 | <.001 | Accepted |
| H3a: FC \rightarrow BI | 0.487 | 0.002 | Accepted |
| H3b: FC \rightarrow EE | 0.879 | <.001 | Accepted |
| H4a: SI \rightarrow BI | 0.600 | <.001 | Accepted |
| H5a: T \rightarrow BI | 0.535 | <.001 | Accepted |
| H5b: T \rightarrow PE | 0.834 | <.001 | Accepted |
| H6a: HM \rightarrow BI | 0.623 | <.001 | Accepted |
| H6b: HM \rightarrow PE | 0.863 | <.001 | Accepted |
| H7a: H \rightarrow BI | 0.816 | 0.001 | Accepted |
| H7b: H \rightarrow EE | 0.384 | 0.583 | Not significant |

B

Fig. 11 The Figure in Section A presents a table showing the mean and standard deviation of the answers to the questions of the UTAUT model. While the Figure in Section B offers a table containing the results of the SEM analysis, in particular indicating which hypotheses were accepted as the final result of the test

considers the chatbot effective (PE), easy to use (EE), well-rated by colleagues (SI), well-supported (FC), trust (T), pleasant (H) and finally the intention to use it in the future (BI). The questionnaire uses a 5-point Likert scale ranging from 1 (strongly disagree) to 5 (strongly agree). Figure 11a reports the mean and the standard deviation of the answers for each question of the constructs.

The average of the responses relating to the "Performance Expectancy" construct is 3.7, therefore between indecision (3) and agreement (4) on the Likert scale. This score can be considered entirely satisfactory, as it indicates that users consider the chatbot a valuable tool to facilitate and speed up the search for LO. The average of the construct "Effort Expectancy" is 4.1. This value suggests that the chatbot is easy to use and that interacting with the chatbot is clear and understandable. The "Social Influence" average is 3.7, while the "Facilitating Conditions" is 4.0. Both values imply that the influence of colleagues is quite relevant and that the user perceives that she/he is well-supported in using the chatbot and has all the necessary knowledge to use it without problems. The "Trust" construct has a resulting mean of 3.5. This value means that people have enough trust in the chatbot's recommendations. The level of trust could grow if more LOs were added to the dataset to return more resources to the teacher's request.

The average score for the "Hedonic Motivation" construct is 3.4, a pretty good result but not optimal. This result may be due to the chatbot interface's limited presence of fun and rewarding components. Unfortunately, RASA restricts the use of various graphical elements, such as animations, which could make the interface pleasant and attractive. The "Habit" construct has a low average of 2.0, indicating that the users do not frequently use chatbots. Therefore, previous experience does not affect the degree of acceptance.

To verify the hypotheses in Fig. 10, we used the structural equation model (SEM) (Fan et al. 2016), combining factor analysis and regression. It first constructs latent variables starting from the items that have been defined and, subsequently, estimates the regressions (specified by the researcher and corresponding to the hypotheses) using the variables above. Through the results of these regressions, it is possible to verify which hypotheses are accepted and with which significance level. As can be seen in Fig. 11b, the SEM model provides an estimated *beta* value and a *p-value* as an output. *Beta* represents the effect of the explanatory variable (the antecedent of the hypothesis) on the dependent variable (the consequent of the hypothesis) and can be either positive or negative. The *p-value* allows us to derive the significance level with which the hypothesis is eventually accepted. In this work, we performed the SEM analysis using Jamovi,³⁴ an open-source tool for data analysis and the realisation of statistical tests.

From the table, it is possible to observe that habit does not influence the construct of Effort Expectancy, as hypothesis H7b has not been accepted. This result means that the created assistant is easily usable even for people who use the chatbot sporadically, like the test participants. The other hypothesis that was not significant is the H2b, which implies the low average obtained for the Behavioral. This value allows us to say the final intention to use the chatbot does not depend on its level of usability.

5.4 Limitations of the experiment validity

We are aware of some limitations that affect our study. Firstly, the sample size of participants in our test. Recruiting only 26 testers does not allow us to present a complete statistical confirmation and validation of the reliability of the collected data, specifically for what concerns the evaluation of the impact of our UTAUT model on the creation of other learning material that is not strictly correlated to programming courses. Secondly, the design of the tests did not focus on specific studies about the combinations of different interactive visualisations that the virtual assistant could use to provide suggestions. Finally, we need to recruit teachers with more expertise in didactics and a more significant number of education domains to evaluate the quality of our solutions. Nevertheless, since all the other hypotheses were confirmed, even with a high beta value, we can claim that our result is an exciting indication because it suggests that the teachers considered the chatbot a helpful assistant even if its usability can be improved.

³⁴ Jamovi. <https://www.jamovi.org/> Last access: 2023-10-26.

6 Conclusion

This paper presents a learning platform that provides teachers with a virtual assistant that helps them create a new digital course. The idea is to use an intelligent assistant to advise teachers about the e-learning modules according to their objectives. It can offer an element of flexibility and customisation of the resources by the teachers to meet their needs. These intelligent suggestions are presented through a visualisation that provides LOs in an accurate, accountable, transparent and well-explained way. The chatbot asks the teacher for the main properties of the course, including the age of the students, the difficulty and the topics covered, necessary to understand the teaching needs of the teacher. Based on the information obtained, the assistant suggests a series of LOs, which the teacher can view and select.

In developing the chatbot, great attention was paid to usability and acceptability, ensuring that the teacher can immediately decide on the LOs and, in the same way, can indicate the data relating to the course. In the design phase, we defined all possible use cases according to which we developed the virtual assistant's actions and the system's general architecture. Subsequently, we moved on to implementing the chatbot through the RASA framework, an open-source framework which, thanks to natural language processing models, allows the creation of sophisticated chatbots. Then, we defined the forms, the RASA components with which the chatbot asks for the user's requested information, and the custom actions necessary to integrate customised functions into RASA.

Our recommendation service takes the course data indicated by the teacher and forwards them to the procedure to facilitate filtering the LOs. For the parsing, we decided to use Sentence-BERT, a machine-learning model based on Transformers, to identify the LOs with the data most semantically similar to the data entered by the teacher. Once we completed its development, the virtual assistant was integrated into an HTML page to be placed in the Google Cloud Platform using Docker and Kubernetes to access the page on the web by IP address.

In the final testing phase, several experiments were conducted on the NLU model to evaluate the chatbot's understanding ability.

To evaluate the impact of the virtual assistant on the teachers' activity, we adopted an extended version of the UTAUT model to study its acceptance and intention to use it. To understand the factors driving the teachers' intention to use the digital assistant's suggestions, we recruited 26 participants. As discussed in the paper, the final results of our tests demonstrate sound effects for concerns about the acceptance of the virtual assistant. Specifically, good values concern the quality of the assistant's ability to communicate effectively, the level of perceived trust in its suggestions and finally, how the teachers' experience affects their perception of the ease of use of the assistant. Further research aims at extending the study involving more teachers with a broader range of competencies in other learning.

Author contributions SV has coordinated the work and written several Sections, particularly the introduction, motivations and related works. He has also followed the valuation of the conversational recommender system. The other authors have focused on research Sections mainly. They have devised recommendation algorithms to suggest the proper LOs and how to combine them. All authors reviewed the manuscript.

Funding Open access funding provided by Università degli Studi di Milano within the CRUI-CARE Agreement.

Declarations

Conflict of interest The authors declare no competing interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Alam, A.: Possibilities and apprehensions in the landscape of artificial intelligence in education. In: 2021 International Conference on Computational Intelligence and Computing Applications (ICCICA). IEEE (2021)
- Alias, S., Sainin, M.S., Soo Fun, T., Daut, N.: Identification of conversational intent pattern using pattern-growth technique for academic chatbot. In: Multi-disciplinary Trends in Artificial Intelligence: 13th International Conference, MIWAI 2019, Kuala Lumpur, Malaysia, November 17–19, 2019, Proceedings 13 (pp. 263–270). Springer International Publishing (2019)
- Angel, J., Aroyehun, S.T., Gelbukh, A.: Nlp-cic@ prelearn: Mastering prerequisites relations, from hand-crafted features to embeddings. (2020). arXiv preprint [arXiv:2011.03760](https://arxiv.org/abs/2011.03760).
- Blei, D., Ng, A., Jordan, M. Latent dirichlet allocation. *Advances in neural information processing systems*, 14 (2001)
- Bocklisch, T., Faulkner, J., Pawlowski, N., Nichol, A.: RASA: open source language understanding and dialogue management. (2017). arXiv preprint [arXiv:1712.05181](https://arxiv.org/abs/1712.05181).
- Campbell, L.M.: Engaging with the learning object economy: Introducing learning objects and the object economy. In: Reusing Online Resources (pp. 53–63). Routledge (2003)
- Chen, L., Chen, P., Lin, Z.: Artificial intelligence in education: a review. *IEEE Access* **8**, 75264–75278 (2020)
- Dagienė, V., Jevsikova, T., Kubilinskienė, S.: An integration of methodological resources into learning object metadata repository. *Informatica* **24**(1), 13–34 (2013)
- De Medio, C., Limongelli, C., Sciarone, F., Temperini, M.: Moodlerec: a recommendation system for creating courses using the moodle e-learning platform. *Comput. Hum. Behav.* **104**, 106168 (2020)
- Deschênes, M.: Recommender systems to support learners' Agency in a learning Context: a systematic review. *Int. J. Educ. Technol. High. Educ.* **17**(1), 50 (2020)
- Devlin, J., Chang, M.W., Lee, K., Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. (2018). arXiv preprint [arXiv:1810.04805](https://arxiv.org/abs/1810.04805).
- Devlin, J., Chang, M.W., Lee, K. Google, KT, Language, AI: BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pp. 4171–4186 (2019)
- Duval, E., Hodgins, W., Sutton, S., Weibel, S.L.: Metadata principles and practicalities. *D-Lib Mag.* **8**(4), 1–10 (2002)
- Fan, Y., Chen, J., Shirkey, G., John, R., Wu, S.R., Park, H., Shao, C.: Applications of structural equation modeling (SEM) in ecological studies: an updated review. *Ecol. Process.* **5**, 1–12 (2016)
- Hoebelheinrich, N., Biernacka, K., Brazas, M., Castro, L.J., Fiore, N., Hellstrom, M., Whyte, A. (2022). Recommendations for a minimal metadata set to aid harmonised discovery of learning resources
- Hwang, G.-J., Xie, H., Wah, B.W., Gašević, D.: Vision, challenges, roles and research issues of artificial intelligence in education (2020)

- Indyk, P., & Motwani, R. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of computing*, pp. 604–613 (1998)
- Khanal, S.S., Prasad, P.W.C., Alsadoon, A., Maag, A.: A systematic review: machine learning based recommendation systems for e-learning. *Educ. Inf. Technol.*, **25**, 2635–2664 (2020)
- Lam, C. S. N., Chan, L. K., & See, C. Y. H. (2018). Converse, connect and consolidate—The development of an artificial intelligence chatbot for health sciences education. In *Frontiers in Medical and Health Sciences Education Conference*. Bau Institute of Medical and Health Sciences Education, Li Ka Shing Faculty of Medicine, The University of Hong Kong.
- Medeiros, R.P., Ramalho, G.L., Falcão, T.P.: A systematic literature review on teaching and learning introductory programming in higher education. *IEEE Trans. Educ.* **62**(2), 77–90 (2018)
- Murad, D.F., Irsan, M., Akhirianto, P.M., Fernando, E., Murad, S.A., Wijaya, M.H.: Learning Support System using chatbot in "Kejar C Package" Homeschooling Program. In: 2019 international conference on information and communications technology (ICOIACT) (pp. 32–37). IEEE (2019)
- Mushtaha, E., Dabous, S.A., Alsyouf, I., Ahmed, A., Abdraboh, N.R.: The challenges and opportunities of online learning and teaching at engineering and theoretical colleges during the pandemic. *Ain Shams Eng. J.* **13**(6), 101770 (2022)
- Okonkwo, C.W., Ade-Ibijola, A.: Chatbots applications in education: a systematic review. *Comput. Educ. Artif. Intell.* **2**, 100033 (2021)
- Palavitsinis, N., Manouselis, N., Sanchez-Alonso, S. (2014). Metadata quality in learning object repositories: a case study. *The Electronic Library*.
- Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S., Carey, T. *Human-computer interaction*. Addison-Wesley Longman Ltd. (1994)
- Reimers, N., Gurevych, I. Sentence-bert: Sentence embeddings using siamese bert-networks. (2019). arXiv preprint [arXiv:1908.10084](https://arxiv.org/abs/1908.10084).
- Roll, I., Wylie, R.: Evolution and revolution in artificial intelligence in education. *Int. J. Artif. Intell. Educ.* **26**, 582–599 (2016)
- Ruiz, J.G., Mintzer, M.J., Issenberg, S.B.: Learning objects in medical education. *Med. Teach.* **28**(7), 599–605 (2006)
- SCORM (2003). *Sharable Courseware Object Reference Model*, Retrieved June 7, 2007, from, <http://www.adlnet.gov/downloads/downloadpage.aspx?ID=243>
- Smutny, P., Schreiberova, P.: Chatbots for learning: a review of educational chatbots for the facebook messenger. *Comput. Educ.* **151**, 103862 (2020)
- Urdaneta-Ponte, M.C., Mendez-Zorrilla, A., Oleagordia-Ruiz, I.: Recommendation systems for education: systematic review. *Electronics* **10**(14), 1611 (2021)
- Valdez, A.C., Zieffle, M., Verbert, K.: HCI for recommender systems: the past, the present and the future. In: *Proceedings of the 10th ACM conference on recommender systems*, pp. 123–126 (2016)
- Valtolina, S., Matamoros, R.A. EUD strategy in the education field for supporting teachers in creating digital courses. In: *End-User Development: 8th International Symposium, IS-EUD 2023, Cagliari, Italy, Jun 6–8, 2023*, to be published in Springer International Publishing (2023)
- Venkatesh, V., Davis, F.D.: A theoretical extension of the technology acceptance model: four longitudinal field studies. *Manage. Sci.* **46**(2), 186–204 (2000)
- Venkatesh, V., Thong, J.Y., Xu, X.: Consumer acceptance and use of information technology: extending the unified theory of acceptance and use of technology. *MIS Quarterly*, pp. 157–178 (2012)
- Venkatesh, V., Morris, M.G., Davis, G.B., Davis, F.D.: User acceptance of information technology: toward a unified view. *MIS Quarterly*, pp 425–478 (2003)
- Warshaw, P.R.: A new model for predicting behavioral intentions: an alternative to Fishbein. *J. Mark. Res.* **17**(2), 153–172 (1980)
- Will, T.: *Measuring and Interpreting System Usability Scale (SUS)*. UIUX Trend (2017)
- Wu, E.H.K., Lin, C.H., Ou, Y.Y., Liu, C.Z., Wang, W.K., Chao, C.Y.: Advantages and constraints of a hybrid model K-12 E-learning assistant chatbot. *IEEE Access* **8**, 77788–77801 (2020)
- Zschocke, T., Beniast, J., Paisley, C., Najjar, J., Duval, E.: The LOM application profile for agricultural learning resources of the CGIAR. *Int. J. Metadata Semant. Ontol.* **4**(1–2), 13–23 (2009)

Stefano Valtolina is an Associate Professor at the Department of Computer Science of the Università degli Studi di Milano, Italy. He obtained a PhD in “Informatics” and an MSc in Computer Science from the Università degli Studi di Milano. His research interests lie at the intersection between End-user Development, Human-Computer Interaction, Interaction and Participatory Design, Human-Work Interaction Design and Computer Semiotics. In this context, he has participated in several national and European Projects in different applicative domains. Specifically, Stefano Valtolina deals with the study and development of methods for the design and evaluation of interactive systems and intelligent conversational agents with the involvement of domain experts in specific application domains (e.g., e-government, Cultural Heritage, Education, Healthcare, SmartCity).

Ricardo Anibal Matamoros is an AI Researcher at Social Things, holding a PhD in Computer Science from the Università degli Studi di Milano-Bicocca. His expertise is primarily focused on developing and evaluating recommendation systems, with a particular interest in Explainability and generative AI. These areas form the core of my research endeavours, where he aims to enhance transparency and user trust in AI systems. His work also involves exploring the potential of generative AI to create innovative solutions for complex problems across various domains, such as digital marketing, personalized content delivery, and interactive media. In addition to technical development, he is deeply involved in the scholarly community, contributing to academic discussions and publishing research findings.

Francesco Epifania Is CEO and founder of Social Things srl (<https://www.socialthingum.it>) and Whoteach (<https://www.whoteach.it/>). He has been a research fellow and teacher at the Department of Computer Science of the University of Milan and received his PhD in Computer Science, together with three degrees in the Computer Science field: a bachelor’s degree in Digital Communication, a master’s degree in Information Technology and a master’s degree in Computer Science; he has produced numerous publications for national and international conferences and journals. His research interests are in Human-Computer Interaction and AI, particularly in creating intelligent, multi-channel interactive systems to enrich knowledge. He is also dedicated to the evaluation of the Recommender System.