Applied Intelligence (2023) 53:24157–24186
https://doi.org/10.1007/s10489-023-04758-3

The AI4Citizen pilot: Pipelining AI-based technologies to support school-work alternation programmes

Athina Georgara[1,2] · Raman Kazhamiakin[3] · Ornella Mich[3] · Alessio Palmero Aprosio[3] · Jean-Christoph Pazzaglia[4] · Juan Antonio Rodríguez Aguilar[1] · Carles Sierra[1]

Accepted: 2 June 2023 / Published online: 17 July 2023

Abstract
The *School-Work Alternation (SWA)* programme was developed (under a European Commission call) to bridge the gaps and establish a well-tuned partnership between education and the job market. This work details the development of the AI4Citizen pilot, an AI software suite designed to support the SWA programme. The AI4Citizen pilot, developed within the H2020 AI4EU project, offers AI tools to automate and enhance the current SWA process. At the same time, the AI4Citizen pilot offers novel tools to support the complex problem of allocating student teams to internship programs, promoting collaborative learning and teamwork skills acquisition. Notably, the AI4Citizen pilot corresponds to a *pipeline* of AI tools, integrating existing and novel technologies. Our exhaustive empirical analysis confirms that the AI4Citizen pilot can alleviate the difficulties of current processes in the SWA, and therefore it is ready for real-world deployment.

Keywords Decision support · AI & Education · Technology pipelining · Natural language Processing · Heuristics & Optimisation

1 Introduction

Citizens daily engage with the public sector in a variety of domains and with different needs. Finding the correct contact and associated procedure, understanding and following the instructions and procedure in the proper manner and making the right decision when several alternatives exist takes much work. Furthermore, conveniently guiding citizens is challenging for public servants, who typically lack decision-support tools to assist them. As a result, the time devoted by public servants to public services makes them costly. Nevertheless, more is needed because the limited availability of public servants to guide citizens hinders the quality of service.

A similar case stands for public services offered in the realm of education, which is the target domain of this paper. In particular, here we focus on *Alternanza scuola lavoro* or School-Work Alternation (SWA) programme leveraging the know-how of *Fondazione Bruno Kessler* (FBK) who is a partner in the organisation of SWA in Trentino. The European Commission calls for reinforcing the partnership between educational institutions and the job market [1]. The aim is to support the development of skills for the employability of new generations, pointed out as a collective responsibility within educational and training contexts. Following European recommendations, Italy started a training programme for students of every high school, for whom it is compulsory to spend a significant number of hours within workplaces during the three last years of their high school course[1]. This scheme is called the School-Work Alternation programme and is described by the Italian Law 107 / 2015: it involves circa 1.5 million pupils ranging from 15 to 19 years old. As argued in [2], the SWA programme creates transformative learning spaces by spurring teachers to investigate how to

✉ Athina Georgara
ageorg@iiia.csic.es

1 Artificial Intellignece Research Institute, CSIC, Campus de la UAB,
Bellatera, Spain

2 Enzyme Advising Group, Passeig de Grácia 17,
Barcelona, Spain

3 Fondazione Bruno Kessler, Via Sommarive 18, Trento, Italy

4 Center of Expertise Public Services, SAP, 805 av du Dr Maurice Donat, Mougins, France

[1] The precise number of hours depends on the schools' typologies.

Fig. 1 The School-Work Alternation programme

approach formal learning to real-world needs and providing students with experiences that help them develop both their technical and soft skills (Fig. 1).

Nowadays, implementing the SWA programme requires much manual intervention from public servants in schools. On the one hand, students are typically interviewed to guide them in their choice of internships on offer. Thus, because of limited resources, providing personalised guidance to students is suboptimal and very time-consuming. On the other hand, schools are challenged with the intricate task of matching internship offers with students' competencies and skills so that they can ultimately assign students to internships while considering students' preferences. Therefore, more decision-support tools need to be devised to aid school personnel in guiding students and producing allocations that satisfy both students and companies offering internships.

Besides that, the current SWA practice individually allocates each student to some internship. This practice hinders the full potential of the programme for two reasons. First, note that team-based, cooperative learning has been shown to tremendously succeed as a learning method in education (as evidenced by e.g. [3–5]). Such a finding has recently spurred research on Artificial Intelligence (AI) algorithms [5–7] and AI-based systems to compose teams in education scenarios (e.g. Team-maker within CATME [8], or Eduteams [9]), as well as on empirical studies (e.g. [7, 10]). Furthermore, since one of the aims of the SWA programme is to promote the development of students' soft skills [2], working in teams would foster the learning of the soft skills that are considered crucial for collaboration and teamwork [11].

In this context, in this paper, we illustrate the development of the AI4Citizen pilot, a software prototype designed as a solution to the shortcomings of the SWA programme. The pilot, developed in the context of the H2020 AI4EU project [12], provides AI tools to automate the SWA programme's current process while introducing new tools to facilitate team-based learning and the acquisition of teamwork skills. Notably, the AI4Citizen pilot involves *pipelining* a range of AI technologies, including (1) NLP algorithms to extract competencies and skills from students' curricula and companies' internship offers and to match them; (2) a chatbot to assist students in selecting internships; and (3) a novel algorithm to group students into teams and allocate them to internships. Specifically, the AI4Citizen pilot tries to make headway in supporting the SWA program with the following contributions:

- *An NLP-based tool to match students with internships.* The main challenge is ensuring that both the employers and the job applicants speak the same language when describing the required skills and competencies. The descriptions are often in free text format, making it difficult to automatically match the job requirements with the applicant's qualifications and experiences. To tackle this issue, we have developed a tool to bridge this gap by utilising ESCO [13], a multilingual classification system of European Skills, Competences, Qualifications, and Occupations. This system encompasses a taxonomy of 13,485 competencies and 13,485 jobs linked with relationships and is designed to map the free text descriptions of the job requirements and candidate experiences to a standardised language.
- *A chatbot to assist students.* In order to develop the AI4Citizen chatbot, we used SAP Conversational AI, an end-to-end collaborative platform for creating chatbots. Our chatbot understands 28 different types of students' intentions related to gathering information about the process and expressing and reviewing preferences. The NLP

engine recognises the intents and triggers suitable skills to interact with the end users. These skills interact with a dedicated component interacting with the database minimising the need to exchange personal information outside the FBK perimeter.

- *An empirical evaluation of the chatbot usability.* We conducted a user study to evaluate our study. Students from three different classes, 55 students in total, participated in the study. During the evaluation, each student was supposed to single out three internships (wishes) and understand the internship context. In the first class, 10% of the students did not manage to complete the evaluation due to a default which thereafter was fixed. The students (90%) who completed the evaluation needed less than half an hour (30 minutes) to do so, and they reported being reasonably satisfied with the experience. Experience satisfaction was improved due to fixes and vocabulary enrichment based on former interactions. However, the main criticism reported was regarding the over-guidance offered by the chatbot. Instead, the users would prefer a free interaction. These findings should be taken into consideration in the future. At the same time, further experiments should be conducted to help us understand the impact on task completion and the time to complete the global mission.
- *An empirical evaluation of the team allocation algorithm.* We conducted a twofold evaluation to confirm the usability of our proposed algorithm. First, we pitched Edu2Com against a state-of-the-art linear programming solver, CPLEX [14]. The results of solving synthetically generated instances of the matching problem showed that Edu2Com outperforms CPLEX in solving time. Second, we tasked Edu2Com to solve large, real-world instances (involving real students' profiles and internship descriptions) of the problem. Our proposed algorithm can handle the problem and find a solution, while an optimal solver such as CPLEX cannot generate the necessary encoding within a reasonable time. Thus Edu2Com solves large problem instances that CPLEX cannot handle.
- *An expert-driven validation of the team allocation algorithm.* Finally, we tasked educational experts with experience in allocating students to internships to assess allocations produced by Edu2Com against allocations manually produced by experienced teachers. The results indicate that our algorithm is the one of choice to solve the problem.

The rest of the paper is organised as follows. Section 2 presents and analyses our case study, Section 3 outlines the overall architecture of the AI4Citizen pilot, Sections 4, 5 and 6 describe the core AI components of AI4Citizen, Section 9 studies the potential business value of AI4Citizen, and Section 7 empirically evaluates the core components of our systems. Finally, Section 10 draws conclusions and sets paths to future research.

2 The *Alternanza Scuola Lavoro* case study

An essential aspect of *innovation* is exploiting information technology to ease people's everyday life. In this work, we followed SAP's best practices and adopted a Design Thinking-led development process (see Fig. 2) to discern users' needs and collect users' feedback.

Scope. As described in the introduction, the AI4Citizen pilot focuses on the SWA scheme. The scheme involves all students from high schools and technical institutes during the last years of their curriculum. The goal of the scheme is to provide practical experience to students, to help them to consolidate the knowledge acquired at school, to test their attitudes on the field, to enrich their training and guide their study path, and, ultimately, to help them to choose their future career path.

Research. In the *research phase*, we focused on understanding how our users interact with the system and their individual needs. To better understand the specifics of this case study, we run a series of interviews with schools' and companies' stakeholders leveraging the central role of FBK in the organisation of SWA in Trentino. We eventually derive the following four personae [15]:

Ludovica she is an 18-year-old student; while attending the fourth class of the classic high school, she is looking for an internship where she can experience team working. She has found a job offer, which seems suitable. However, she does not exactly understand which skills the company requests—Fig. 3 shows a detailed view of Ludovica's persona.

Arnoldo he is a teacher 61 years old; he manages the SWA office at his Institute, sharing this duty with a colleague; they help more than 300 students every year first to find out a suitable internship; then, they follow the initial administrative work, resolve problems during the internship, and at the end, collect students' final report and tutors' evaluation form.

Carolina she is a 48-year-old assistant in the internships office at a large research centre; she takes care of the relationships with local high schools, interacts with tutors like Arnoldo when they have to select a team of interns, and chases researchers to stimulate them to prepare new internship offers. Carolina also supports researchers in preparing the final evaluation reports.

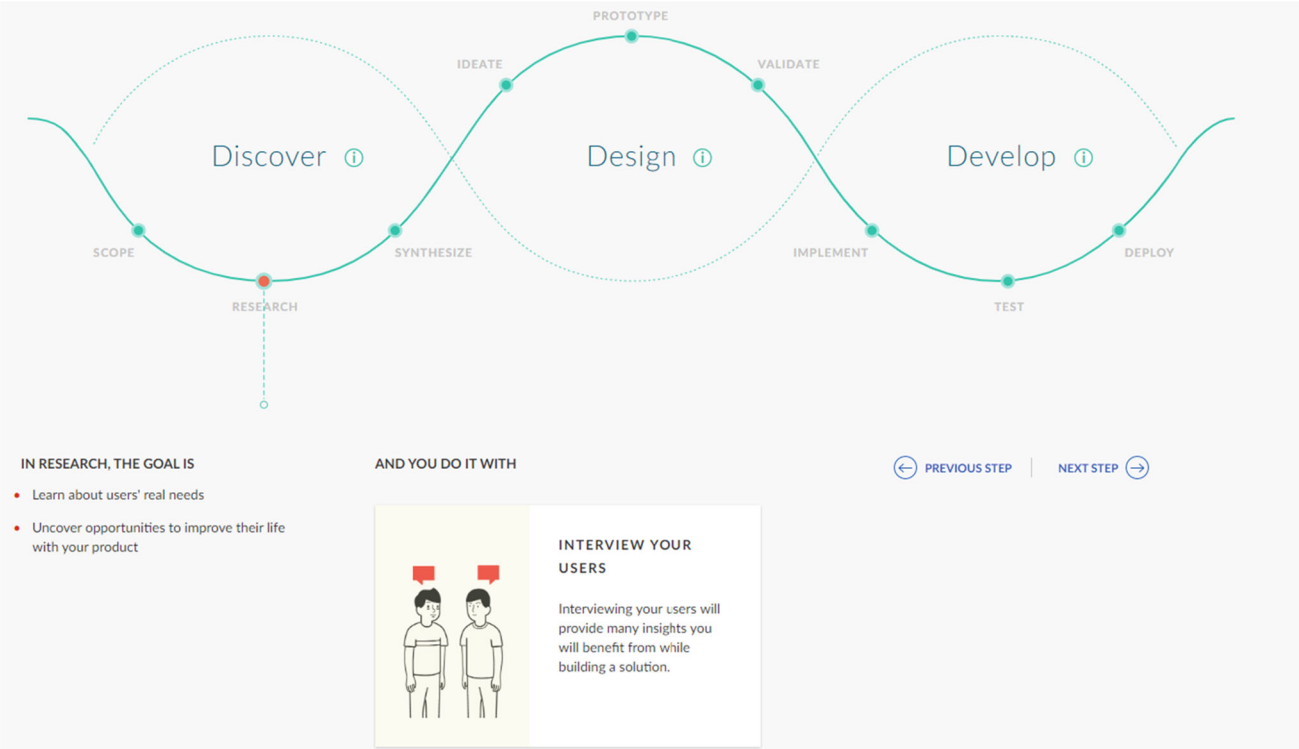


Fig. 2 Design led approach (from https://saptraining.build.me/)

Rosanna she is a 37 years old busy researcher; every summer, she looks for a team of two to four students able to help her with pet projects, also aiming to transmit her passion for scientific research work.

Synthesize. Following the interviews with teachers, researchers, and parents, we focused on Ludovica's and Arnoldo's experiences to *synthesize* the main scenario identifying their main pain points. The following paragraph, the *scenario as-is*, describes the typical interaction between our student Ludovica and the reference teacher Arnoldo nowadays.

> Arnoldo is the reference teacher for the SWA for a set of classes. During the past weeks, he has already updated a list of all the internship offers received, classified, and published it on the school's website. He has also sent several emails to all the students and their parents during the previous month concerning new internship proposals. First, Ludovica checks her mailing box to collect all these emails and then scrolls the internship list on the school website. She also asks a friend of her family, who owns a high-tech company, if he can host her for an internship. He answered positively to Ludovica. After this work, she prepares a list where she writes the internship proposals that seem interesting to her. In this list are two internships: an internship extracted from one of the emails sent by the reference teacher and the internship in the company of a friend of Ludovica's family. At this point, Ludovica would like to discuss the two different options with Arnoldo, the teacher who manages SWA at her school. Ludovica should make an appointment with him. They set up a meeting during which Ludovica would like to interact with Arnoldo to understand the internship goals and objectives better and determine if the family friend internship can be performed. The discussion should focus on understanding how the different internships match Ludovica's competencies, objectives, weaknesses, and strengths. Unfortunately, Arnoldo has to manage thousands of questions from students and can dedicate only a fraction of the needed time to tackle the different questions, which eludes the family's friend's internship question. Ludovica feels totally lost in the process and fills out a form with three preferences. A few weeks later, while Arnoldo is trying to make an educated choice after receiving the preferences of all students, composing the teams for the different internships looks to him like solving a complex jigsaw but without knowing the image to compose! Still, Arnoldo remembers the discussion with Ludovica, and he finds out that an apparently *good* global solution enables her to get assigned to her second choice. Apparently, the young Ludovica is frustrated,

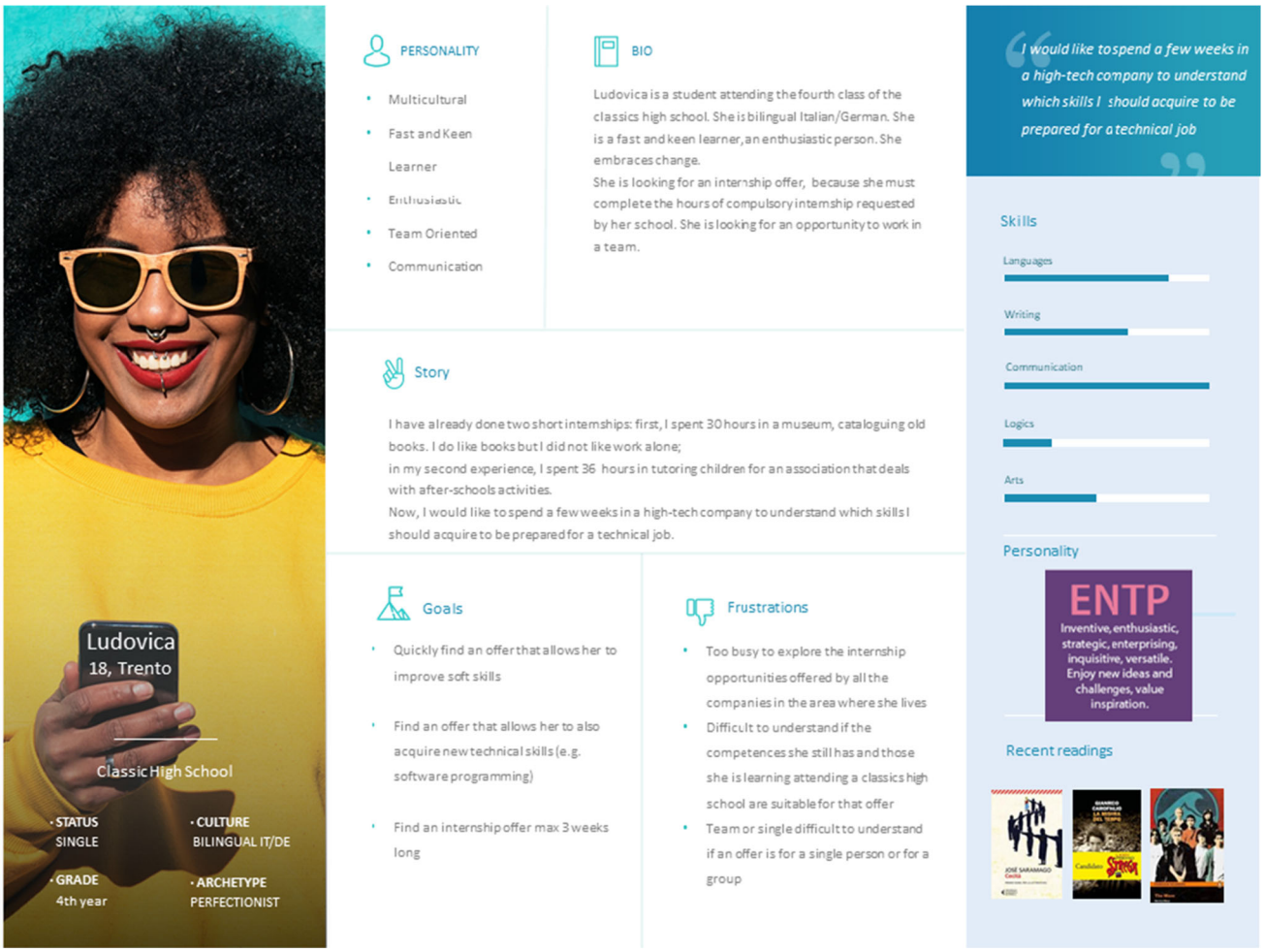


Fig. 3 The Ludovica persona

especially when Arnoldo cannot precisely explain how he figures out these internships' matchings fighting with his spreadsheet.

Following this return on experience from the current process, and in order to enhance the student and referent experience, we decided to focus on two specific problems, which are particularly time-consuming and not supported by the current version of the *Vivoscuola* portal[2] operated by the Autonomous Province of Trento:

1. How to best inform and guide Ludovica in her internship quest with a teenager-ready conversational interface?
2. How to support Arnoldo in building teams of complementary interns while optimizing the acquisition of new skills and their personal wishes?

Ideate. The following descriptions, the *scenario to-be*, explain how AI will modify Ludovica's quest for the perfect internship and support Arnoldo to dedicate more quality time.

An announcement was made on the school billboard, informing students that it was time to start thinking about their SWA options. Students were informed that they could use social media channels like Facebook and Twitter to search for and enrol in internships. These channels were securely connected to the SWA-FBK system, allowing it to access information about each student, including their competencies, previous internship experiences, and past year statistics and testimonials. Ludovica decided to connect to the chatbot using her SWA-FBK credentials. The general chatbot asked her questions to determine her interests and preferences, using both the information stored in the SWA-FBK system and additional information provided by Ludovica. The chatbot provided Ludovica with answers to common questions. At the same time, it redirected some questions about a specific intern-

[2] For more information visit https://www.vivoscuola.it/

ship offer to Arnoldo, who receives a small portion of questions so he is able to answer in due time. The chatbot proposed several internship programs aligned with Ludovica's interests and preferences and passed this information on to the team formation algorithm. After a couple of hours, Arnoldo received the team assignment provided by the algorithm. Although Ludovica had expressed a strong desire to be assigned to her first choice, Arnoldo realized that this constraint decreased the overall assignment score and the fit of five teams, including Ludovica's. As a result, he followed the algorithm's recommendation and assigned Ludovica to the internship that was her second choice, explaining the rationale behind the decision to her.

3 A blueprint of the AI4Citizen architecture

Before proceeding with the implementation details of each component, we detail in this section the overall architecture of the AI4Citizen pilot. The architecture of our pilot is funded on pipelining the different intelligence technologies: the competence and skill extraction tool, the chatbot assistant and the team formation algorithm. At the same time, these technologies are to be integrated with FBK's existing IT system that manages students' profiles (information regarding their studies and past activities) along with internship program offers. As a result, we deliver a prototype implementation of our end-to-end AI4Citizens pilot. In a nutshell, Fig. 4 illustrates the pilot's architecture, which consists of the entities below:

- A module for interfacing with and integrating from the external — preexisting — an IT system that contains and manages the information regarding students' profiles, their activities, and experiences, etc.
- A module for interfacing with and integrating from the external IT system that contains and manages job/internship offers information.
- Data management services to facilitate the "normalization" of the data in terms of vocabularies and taxonomies for the purpose of the offer/candidate matching. More specifically, the two key services here are *(i)* skill matching to associate an entity with the skills and competences in the ESCO ontology [13]; and *(ii)* a multi-dimensional classification of internships to guide the selection process. The latter service characterizes offers in different practical ways (e.g. activity domain, geographical distribution, context (*e.g.,* private vs public hosting entities), etc.).
- A **team formation** service to match teams of students to internships taking into account competencies, preferences, and availability in a holistic, cross-organizational manner.
- An **internship browser** that brings this information together, exposing different APIs for searching, match-

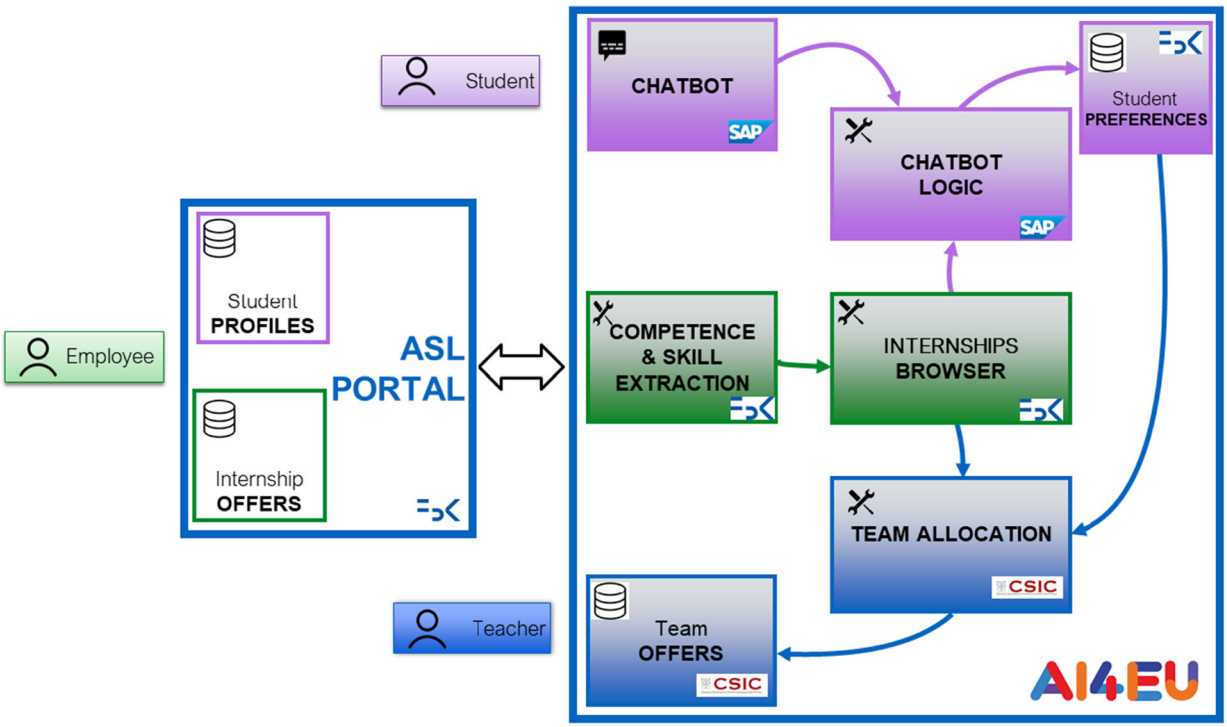


Fig. 4 Components and use cases dataflow

ing, and selecting offers, as well as for storing preferences and matching teams to available offers.

- A **chatbot-based Natural Language Processing (NLP) and User Interaction (UI)** service and its associated **chatbot logic** component that make the AI4Citizen pilot available to the main stakeholders, in particular to students and teachers. While the role of the UI is to make AI4Citizen directly available on the web or a cell phone, the chatbot logic drives the selection process, answers frequently asked questions and helps collect students' preferences.

The architecture of the AI4Citizen pilot considers two execution environments. On the one hand, the *SWA FBK* environment runs the SWA system prior to AI4Citizen. It is in charge of storing and handling the schools' and companies' data. On the other hand, the AI4EU environment hosts the AI4EU services required to enable an AI-based SWA system, namely competence and skill extraction, team formation, the chatbot, and the internship browser. The two environments, *SWA FBK* and *AI4EU*, provide strong data encapsulation, limiting the interactions with the new components to a set of restrictive APIs to avoid the usage of any Personal Identifiable Information and to respect the term of services between FBK and the Trento province.

Our implementation follows the principles of Cloud-Native Applications. It uses a container-based component model, where each separate micro-service is deployed in a Docker container on top of the Cloud infrastructure. The Chatbot NLP and UI component run on the SAP cloud and interact with the SAP logic thanks to a REST API over a secure channel.

The following sections propose a detailed description of the main components (Table 1).

4 Competence and skill extraction

Every day thousands of people look for a job and share their curriculum vitae using different channels, like social networks, leasing companies, etc. In the meantime, a similar number of companies look for workers and use the same channels to share the requirements for their open positions. In the context of the scenario presented in this paper, a similar situation arises between the students looking for internships and the companies searching for candidates with appropriate skills and competencies.

The fundamental problem here is to make the two parties speak the same language when it comes to characterising the skills and competencies required by the companies and those owned by the candidates obtained through the activities and experiences they have carried out during their careers and/or studies. Indeed, frequently the descriptions of experiences and school curriculum are expressed in natural language, making it challenging to match requests and offers in an automated manner.

To address this issue, we have defined and developed a tool that bridges the gap between the required and provided competencies when expressed as a free text description.

To reach this goal, we use two existing resources, *(i)* the ESCO Ontology [13] and *(ii)* the FastText software [16]. ESCO is the multilingual classification of European Skills, Competences, Qualifications and Occupations. It contains a taxonomy of 13,485 competencies and 13,485 jobs connected with relations. It has been developed as a part of the Europe 2020 strategy. It identifies and categorises skills, competencies, qualifications, and occupations relevant to the EU labour market and education and training. It systematically shows the relationships between the different concepts described in 27 European languages. It has been recently adopted in various countries and regions to describe the

Table 1 The AI4Citizen pilot components published on the AI4EU platform

Asset name	Catalogue	Link
Chatbot (SAP Conversational AI)	AI Catalogue	https://www.ai4europe.eu/research/ai-catalog/sap-conversational-ai
Team Formation	AI Catalogue	https://www.ai4europe.eu/research/ai-catalog/edu2comapi
(edu2comAPI)	AI Exp	https://aiexp.ai4europe.eu/#/marketSolutions?solutionId=2e1ac521-e09b-4451-81f6-3f8d107b98da&revisionId=505a114b-edbf-4101-b007-4fa09afa4a62&parentUrl=marketplace
Competences & Skills Extraction (ai4eu-competences)	AI Exp	https://aiexp.ai4europe.eu/#/marketSolutions?solutionId=dc67374a-0a1c-4477-86b2-9db8f0a1faed&revisionId=977872e8-b343-4fa4-b5fe-31afc77c9e05&parentUrl=marketplace

professional context in the corresponding territories unambiguously. We use the ESCO ontology as a basis for the description of the job and internship offers as well as for characterising candidates' skills and experiences.

The latter resource, FastText, is a state-of-the-art technology developed by Facebook, which computes the semantic similarity between two text spans. It is released with an open-source license and is available in over 100 languages.

FastText is an evolution of word2vec [17], a word representation model where each word is an object in a vector space, and its position is optimised for the task of predicting the surrounding context.

The main difference between the two models is that the representation v_w of a word w is not only the representation of its symbol, but it is augmented with the sum of the representations of its subword units:

$$v_w = u_w + \sum_{s \in \mathcal{S}} u_s$$

where $\mathcal{S}$ is the set containing some character n-grams contained in w, u_w is the vector representation of the whole word w and u_s are the representations of the subwords.

Combining the two resources, we have developed the so-called *Competence and Skill Extraction* (CSE) tool, an AI component that provides the following functionalities:

- using FastText, we compute the semantic similarity between plain text (from a job description or CV experience) and the competencies appearing in the ESCO ontology;
- we then rank the identified competencies using this similarity value; and
- according to the extracted ranking, we suggest the most appropriate candidate competencies for the description.

As a positive side effect of using native multi-language resources, the tool is automatically able to compare texts written in different languages. For example, the text "application of rules and scientific methods to solve problems" can be semantically connected to the competencies "develop strategy to solve problems", "plan activities to accomplish specific goals", and "problem solving". A similar result can be obtained by replacing the English text with the Italian one, "applicare regole e metodi scientifici per risolvere problemi". Consequently, the tool may be used in international markets, making it possible to relate different IT systems and their data across EU boundaries and facilitate internships abroad in our context.

It is important to remark that the tool allows for a wide range of use cases in the context of the proposed scenario and broader settings. In particular, the tool allows to:

- annotate existing job descriptions with the skill/competence information for further elaboration;
- guide the description and annotation of the information within the IT systems managing the corresponding data (such a tool may assist the operators in appropriate and concise data entry activity);
- support the data analysis tools when reasoning across all the data, also considering historical perspective, legacy sources and data, etc.
- bridge the gap between the systems using different skill/competence taxonomies in different contexts, such as different territories and organisations.

The implementation of the skill matching tool is done in Python and is available on GitHub under the Apache 2.0 license [18].

5 Chatbot

To enable a smooth interaction between students and the system so that students can learn about the SWA program, get to know the offered internship programs, and ultimately express their preferences, we early took the decision to leverage the concepts of the conversational user interface. Conversational UX is becoming a *de facto* standard to address processes of medium complexity that can be customised to a specific user by gathering contextual information (*e.g.,* student profile, topics, grade, past internships, past conversation). Leveraging this information, we enable students to ask questions using natural language about the process itself in a generic (*e.g.,* What is an SWA program?), contextual (*e.g.,* Who is my referent professor?) or individualised (*e.g.,* What are my recommended internships?) way, which will eventually trigger questions to propose a pertinent answer. The choice of a conversational UX also enables to leverage the inclination of Generation Z better to use instant messaging platforms.

Our choice to use SAP Conversational AI [19] to implement the AI4Citizen chatbot was due to different factors:

1. It does not provide self-learning capabilities since we wanted to avoid the danger of bot manipulation (*e.g.,* the Tay experiment [20]);
2. Its NLP engine provides multi-language support, including Italian, Spanish and Catalan on top of English;
3. It mixed no-code and scripting capabilities with an API, enabling interaction in a bidirectional way with other software components;
4. Its platform provides enterprise-grade level including GDPR support, monitoring and debugging capabilities.

To better understand a chatbot's development process, we need to introduce some terminology used in the domain.

The main aim of a bot is to understand the *expression* that a user might say. For this purpose, the NLP engine of the bot will analyse the expression and extract features such as date, city or person names - the *entities* - and enrich them with information suitable for programming (*e.g.,* the GPS location associated with a city) and eventually the *sentiment* analysis of the expression. Twenty-eight types of entities, the *gold entities*, are predefined, and the programmer can add custom entities to adapt the capability of the chatbot to a given domain. Once this first stage is performed, the conversational engine attempts to match the expression — thanks to a concept of expression distance — with one of the expressions that are captured at the design stage to express an *intent*: intents are the core of the chatbot and are responsible for understanding user requests. Intent recognition is leveraging our platform's machine-learning capabilities based on the corpus of expressions defined in the intent. To answer the intent, a *skill* is triggered that will eventually ask the user to enrich its request with details represented by entities (*e.g.,* the time of departure for booking a train to a city) to enable to propose a pertinent answer and continue the conversation. *Action* enables skill to build the answer directly on the platform with *message* or interact with external components with *webhook* or trigger complementary skills.

The SWA chatbot leverages all these capabilities to interact with the student on one side and with the information stored about their curriculum or previous session. We can divide the skills developed into four groups:

- 14 skills related to answering frequently asked questions
- 10 customised skills that capture preferences and provide individual advice to the student
- 2 skills that implement an authentication pattern to interact with the customised skills
- 2 skills developed for facilitating the testing sessions

These skills are associated with circa 30 intents capturing more than 200 expressions. We also enriched the gold entities with eight custom entities serving three different purposes:

- to enable domain-specific interactions, this encompasses competencies (*e.g.,* related to the ECSO ontology), companies' names and cities contained in the internship database. While we initially did not foresee adding cities as entities, we realised that the related gold entities lack precision for small cities. Fortunately, these entities can be loaded thanks to the chatbot API automatically upon database modification;
- to capture the format of specific identifiers;
- to enable the students to refer to the chatbot artefacts (*e.g.,* to ask questions about the chatbot itself).

In order to propose individual advice to students, the system should access or store information related to each student.

This access should be done securely, and due to privacy concerns, we had to minimise the information stored at the chatbot level. For that purpose, our chatbot relies on a set of *webhook*, which uses a stateless proxy, called *chatbot logic*, residing on the FBK realm. The communication is done via a secure channel where we transit the information captured during the conversation with the student. The chatbot logic interacts with the internship browser component to store or retrieve the information in order to build the answer based

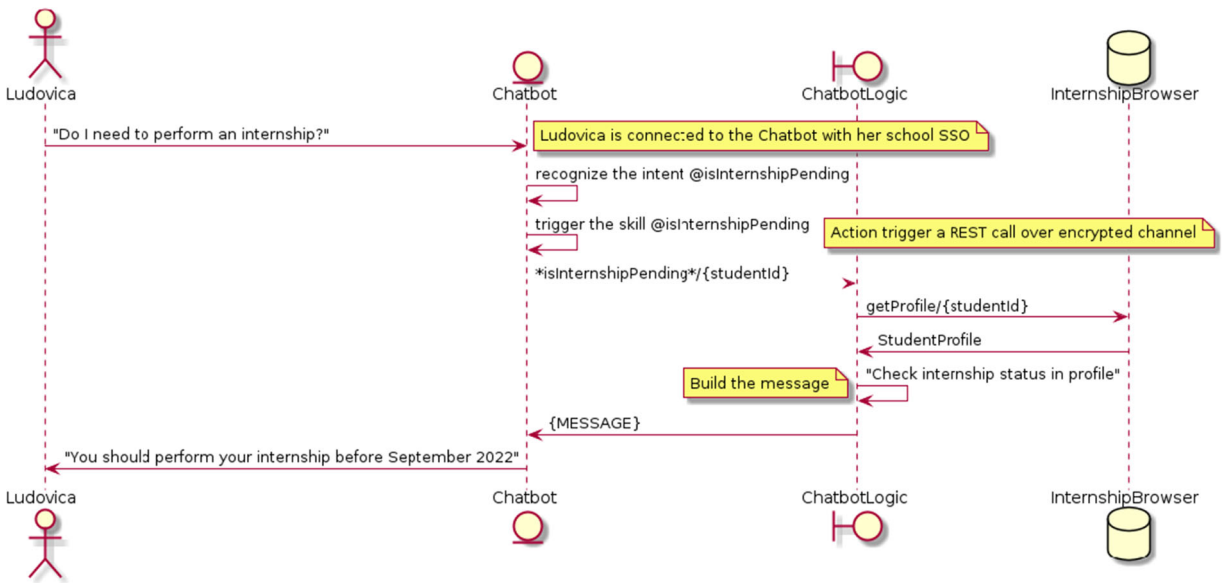


Fig. 5 Sequence diagram showing a simple interaction

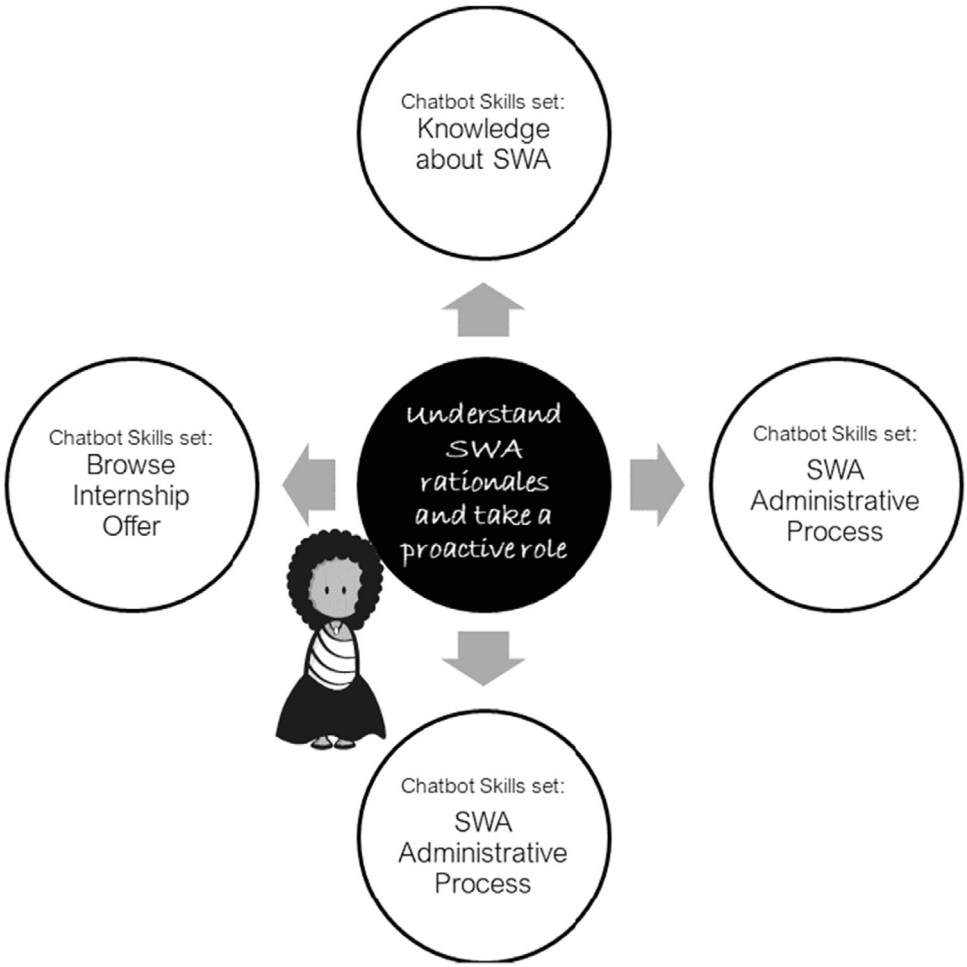


Fig. 6 Ludovica and the chatbot skills set

on *(i)* the academic student profile (*e.g.,* schools, grade, curriculum), *(ii)* the preferences already captured in the current session or previously, and *(iii)* the information captured by the current skill. Aside from the privacy advantages of such an approach, it also enables us to benefit from a full-fledged programming language to elaborate the answers relying on the provided message library (*e.g.,* text, card, button). Figure 5 shows an example of a simple interaction during the early stage of the conversation.

To sum up, Fig. 6 shows the different type of skills and their related actions that enables Ludovica to engage with the chatbot in order to better understand the rationales around the SWA activity, to ask questions about the process and administrative details, to express preferences, and to browse internship offers.

6 Allocating teams of students to internships

This section focuses on the problem of forming teams of students and matching them with internships (i.e., allocating student teams to internships), which is equivalent to the team formation service shown in Fig. 4. Due to the complexity of this problem, it is difficult and time-consuming for an expert to make optimal allocations. Previously, we used existing technologies to develop the competence and skill extraction service and the chatbot, as explained in Sections 4 and 5. However, creating our team formation service requires new research contributions. Section 6.1 explains why this is necessary, followed by a formal definition of the allocation problem in Section 6.2. Then, in Section 6.3, we provide the encoding for the allocation problem to be optimally solved as a linear program (LP) with some state-of-the-art existing solver. Lastly, we introduce a novel heuristic algorithm for computing team allocations in Section 6.4. This algorithm allows us to tackle real-world problems of significant size, which cannot be solved with sophisticated optimisation libraries like CPLEX, as demonstrated in Section 7.3.

6.1 Motivation

Many real-world problems require teams of people or machines to work together on tasks, such as robot teams

in search and rescue missions [21] or teams of drones for surveillance [22]. As a result, there has been much research on various methods for assigning teams to tasks in the field of artificial intelligence.

Existing literature has explored how to form *one* team to work on *one* task [23–25], form *one* team to work on *many* tasks [26], and form multiple teams to solve a single task [5, 10, 27, 28]. Some research has also been done on forming multiple teams to match multiple tasks, allowing overlaps either on agents participating in many teams [21] or on many teams performing the same task [29]. However, there needs to be more attention given to the problem of how to allocate teams to tasks with no overlaps, i.e., teams that share no common members and tasks that are uniquely tackled by one team. This problem above is the one our team formation service aims to address when allocating teams of students to internships. Despite limited existing research on this topic [30, 31], we cannot rely on prior approaches to solve our problem due to several shortcomings. Specifically, one of the approaches uses brute force and branch-and-bound techniques that limit the number of agents and tasks that can be analyzed [31], while the other approach handles only a small number of tasks [30].

Additionally, existing models assume that a team must possess the exact competencies a task requires. Such an assumption is only sometimes practical in real-world scenarios. As [28] points out, existing literature endures this assumption and distinguishes between two competence models: *boolean* models where an agent acquires or lacks some competence [23–25, 30], and *graded* models where an agent acquires some competence up to some degree [5, 27, 28, 32]. However, in the real world, it might be the case that acquiring some *similar* competence is sufficient for handling a specific required competence. Consider, for example, the educational world, where a student can be adequate for some internship program even if the student does not possess all the required competencies precisely as requested as long as the student's competencies are similar enough.

Therefore, this work aims to develop a new formalization and algorithm that can practically address the many teams to many tasks allocation problem. In what follows, we use the terms agents and tasks instead of students and internship programs to present the problem more broadly. Later (in Section 7.3), we will apply our general problem to our case study of matching teams of students with internship programs. This approach will help us consider the competencies possessed by team members, including similar competencies that may still qualify a student for an internship, even if they do not possess the exact competency required by the task. Ultimately, our goal is to develop a novel algorithm that can effectively allocate teams of agents to tasks in real-world scenarios.

6.2 Defining the team allocation problem

This section aims to cast the problem as an optimisation one formally. First, we introduce the basic concepts of the team allocation problem, then we move to the concept of competence models and put forward the model used in this work. Finally, we formally define the optimisation problem.

6.2.1 Basic Concepts

According to the Oxford Learnner's Dictionary[3], *competence* refers to the ability to perform well, the authority or power in handling a particular situation, or a necessary skill for performing a particular job or task. Let us denote with $\mathcal{C}$ a fixed set of competencies. A task is described through its requirements on *(i)* competencies and *(ii)* team size. Firstly, for some agent(s) to successfully carry out a task, they must adequately deal with the required competencies. Secondly, the required team size signifies the necessary and sufficient number of agents working together to complete the task successfully—i.e., a team with fewer members than the required team size will not be able to complete the task, while a larger team over-consumes (agent) resources. For example, an internship program in a computer tech company might require four competencies (machine learning principles, coding in Python, web development, and fluency in English) and a team of size three. Therefore, the company offers an internship that needs three students together who possess the four required competencies. Moreover, even though every required competence is necessary for successfully completing a task, not all competencies are equally important; as such, we consider the competencies' relative importance as part of the task description. For example, for an internship in a computer tech company, coding in Python might be more critical than fluency in English. Formally, *taask* τ is given by $\langle C_\tau, w_\tau, \mathrm{s}_\tau \rangle$, where $C_\tau \subseteq \mathcal{C}$ denotes the required competencies, $w_\tau : C_\tau \to (0, 1]$ is a relative importance weight function, and $\mathrm{s}_\tau = 1, 2, \cdots$ denotes the required team size.[4] We denote with $T = \{\tau_1, \cdots, \tau_m\}$ a set of m tasks (with $|T| = m$). An agent is described through the competencies they possess. A *team* K corresponds to a subset of agents, $K \subseteq A$, who jointly work on some task(s). A size-compliant team K for some task $\tau \in T$ is a team such that $|K| = \mathrm{s}_\tau$, i.e., a team with as many members as required by τ. We denote with $\mathcal{K}_\tau = \{K \subseteq A : |K| = \mathrm{s}_\tau\}$ the set of the size-compliant teams for τ given a set of agents A.

[3] https://www.oxfordlearnersdictionaries.com/

[4] Note: we use subscript τ to refer to the required competencies, relative importance, and team size of task $\tau \in T$. Similarly we will use subscript a to distinguish the elements describing an agent $a \in A$.

6.2.2 Competence coverage and affinity

In order to assign a team of agents to a task, it is necessary to ensure that the team possesses the competencies required to solve the task. To achieve this, the competencies, as determined by C_τ, must be assessed before allocating a team to a task. For a team of agents K to be considered suitable for a task τ, the team must possess the necessary competencies required by τ. That means that for each required competence by the task, there must be at least one agent in the team who possesses that competence.

The current literature on team-task allocation considers competencies as either Boolean or graded features, with a team's matching quality being determined by a function, usually expressed as a utility function. However, these existing models are pretty restrictive, so a new approach is needed. In this study, we propose a method for assessing the matching quality of a team with a task based on the semantic similarity of the competencies required by the task and the competencies possessed by the team. To be more precise, we introduce the concepts of *competence coverage* and *competence affinity*, which provide an intuitive way to determine the matching quality of a team for a task. Competence coverage is determined by the degree of *similarity* between the required competencies of the task and the team's collective competencies. Considering the *semantic similarity* of the competencies allows us to deal with the current competence models' limitations.

Within the scope of some specific domain (e.g., the educational domain), we can structure the competencies related to the domain to capture semantic relations among them. More and more entities (countries, organisations, institutes, etc.) work towards registering and structuring competencies in ontologies—e.g., ESCO [13], O*Net [33], SFIA [34], ISFOL [35]. For example, two essentially different competencies, such as coding in C++ (competence c) and coding in Java (competence c'), share essential principles, e.g., both are compiled languages and object-oriented languages.

Therefore, we assume that the competencies in $\mathcal{C}$ are structured in a competence ontology based on the semantic relations among the competencies. Then, for any two competencies $c, c' \in \mathcal{C}$, there is a degree of similarity, denoted as $\text{sim}(c, c') \in [0, 1]$, reflected in the competence ontology. Moreover, we assume that the competencies can be represented in a (tree-like) graph structure, where nodes denote competencies and directed edges indicate intimate semantic relations between the nodes (such that Parent nodes are broader concepts of their successor nodes). In Section 7.3, we utilise ESCO ontology, a well-established ontology with such properties. Given a competence ontology of $\mathcal{C}$ and two competencies $c, c' \in \mathcal{C}$, to compute the semantic similarity between c and c' we use a variant of the similarity metric presented in [36]. Unlike [36], the variant we propose here guarantees that any competence node is maximally similar to itself (i.e., $\text{sim}(c, c) = 1$). Thus, the similarity is given by

$$\text{sim}(c, c') = \begin{cases} 1 & \text{if } l = 0 \\ e^{-\lambda l} \frac{e^{\kappa h} - e^{-\kappa h}}{e^{\kappa h} + e^{-\kappa h}} & \text{if } l > 0 \end{cases}$$

where l stands for the shortest path connecting competencies c and c' and h stands for the depth of the deepest competence subsuming c and c'. Parameters κ, λ control the impact of the distance l and depth of common ancestor h on the similarity metric, respectively. Similarly to [37], for any two $c, c' \in \mathcal{C}$ it holds that $\text{sim}(c, c') \in [0, 1]$.

We say that some agent $a \in A$ with acquired competencies $C_a \subseteq \mathcal{C}$ covers competence $c \in \mathcal{C}$ with a's most similar competence c', and specifically with a degree reflecting the semantic similarity between c and c', i.e., $\text{cvg}(c, a) = \max_{c' \in C_a} \text{sim}(c, c')$. Moreover, we say that an agent $a \in A$ covers a task $\tau \in T$ with required competencies $C_\tau \in \mathcal{C}$ with a degree depending on how well a can cover each of the required competencies. Specifically, we use the product over a's competence coverage across all required competencies:

$$\text{cvg}(a, C_\tau) = \prod_{c \in C_\tau} \text{cvg}(c, a) = \prod_{c \in C_\tau} \max_{c' \in C_a} \{\text{sim}(c, c')\} \quad (1)$$

In a team context, it is unlikely that each agent will need to cover all the required competencies. Instead, team members collaborate and complement each other by sharing responsibilities [38]—notably, [39] highlights that recent studies define team through assigning responsibilities to agents. Thus, we must solve a *competence assignment function*. Let $\tau \in T$ be a task and $K \subseteq A$ be a subset of agents that forms a team; each agent, $a \in K$, is assigned with some of the required competencies in C_τ. That is, each agent in the team is responsible for covering only *some* of the competencies required by the task. Therefore, a *competence assignment function (CAF)*, denoted as $\eta_{\tau \to K} : K \to 2^{C_\tau}$, assigns a subset of required competencies to each team member $a \in K$. As noted in [27, 28], a CAF should satisfy the following property: at least one agent, $a \in K$, must cover any competence $c \in C_\tau$, i.e., $\bigcup_{a \in K} \eta_{\tau \to K}(a) = C_\tau$. Additionally, let the *reversed* function (r-CAF), denoted as $\theta_{\tau \to K} : C_t au \to 2^K$, show which agents are responsible for each required competence. Let $\Theta_{\tau \to K}$ be the set of all CAFs of some team K for task τ. Notably, there are many different CAFs, though not all CAFs are equally good, i.e., some CAFs are better than others. [27] discusses *inclusive* competence assignments, thus, here we build on inclusive CAFs, and we propose the *fair competence assignment functions (FCAF)*. In more detail, an FCAF satisfies two properties: each agent is responsible for *(i)* at least one competence (as in inclusive competence assignments), and *(ii)* at most $\left\lceil \frac{|C_\tau|}{|K|} \right\rceil$. These properties ensure

that all agents actively participate in the teamwork and that a few agents are assigned excessively many responsibilities while others are assigned just the minimum.

Given a task τ, a team K and an FCAF $\eta_{\tau\to K}$, we assess the matching quality of K working on τ as how well each team member can cover their responsibilities according to the FCAF at hand. Thus, we use the *competence affinity* metric, which *(i)* awards high coverage of competencies and *(ii)* does not penalise low coverage of competencies with low relative importance. First, we define the competence affinity from a single agent's perspective:

Definition 6.1 (Agents' Competence Affinity). Given an agent $a \in A$, a task $\tau \in T$, and a competence assignment function $\eta_{\tau\to K}$, the competence affinity of a to τ is:

$$\mathrm{aff}(a, \tau, \eta_{\tau\to K}) = \prod_{c\in\eta_{\tau\to K}(a)} \max\left\{\left(1 - w_\tau(c)\right), \mathrm{cvg}(c, a)\right\}. \tag{2}$$

Then, the team's competence affinity depends on the competence affinity of each team member. Specifically, it is defined as the *a-la-Nash* product over the individual competence affinity of every agent in K. The product promotes equal contributions across the agents in the team (i.e., all agents are equally contributing to the task) and therefore favours teams with balanced contributions. Formally, the competence affinity from a team's perspective is defined as:

Definition 6.2 (Team's Competence Affinity). Given a team of agents $K \subseteq A$, a task $\tau \in T$, and a fair competence assignment $\eta_{\tau\to K}$, the competence affinity of K to τ is:

$$\mathrm{aff}(K, \tau, \eta_{\tau\to K}) = \prod_{a\in K} \mathrm{aff}(a, \tau, \eta_{\tau\to K}). \tag{3}$$

Notice that a team's competence affinity changes as responsibilities are assigned differently, i.e., the very same team exhibits different matching qualities when different FCAFs are in place. The FCAF that results in the best competence affinity of a team for a task amounts to solving an optimisation problem:

$$\eta^*_{\tau\to K} = \arg\max_{\eta\in\Theta_{\tau\to K}} \mathrm{aff}(K, \tau, \eta) \tag{4}$$

such that $\bigcup_{a\in K} \eta^*_{\tau\to K}(a) = C_\tau$ and $1 \le |\eta^*_{\tau\to K}(a)| \le \left\lceil \frac{|C_\tau|}{|K|} \right\rceil$ for every $a \in K$. In practice, though, we can efficiently solve the above optimisation problem optimally since team sizes and the number of required competencies for a task (and therefore the search space $\Theta_{\tau\to K}$) are relatively small—usually team size ranges in [2, 5] and a task requires ≤ 10 competencies.

Then, for task τ, considering all the size-compliant tams $\mathcal{K}_\tau$, the best team shall maximise the competence affinity; i.e., $K^* = \arg\max_{K\in\mathcal{K}_\tau} \mathrm{aff}(K, \tau, \eta^*_{\tau\to K})$. Note that the search space for this optimisation problem is $\binom{n}{s_\tau}$ large (and increases as n and s_τ increase), where n is the number of agents in A and s_τ is the team size of task τ.

6.2.3 The optimisation problem

Now, given a set of tasks T (with $|T| > 1$), finding a team for each task, i.e., computing a many teams to many tasks allocation, is another optimisation problem. In more detail, the optimal allocation of agent teams to tasks is the one that maximises the competence affinity of every team, respecting *(i)* that each agent cannot be part of more than two teams; *(ii)* each team can work on at most one task; and *(iii)* at most one team can work on each task. These three properties we just mentioned describe a *Feasible Team Allocation Function*, which we formally define as:

Definition 6.3 (Feasible Team Allocation Function (FTAF)). Given a set of tasks T, and a set of agents A, a feasible team allocation function g is a function $g : T \to 2^A$ such that: (1) every task $\tau \in T$ is allocated its requested number of agents so that $|g(\tau)| = s_\tau$; and (2) an agent can only be assigned to one team: for every pair of tasks $\tau, \tau' \in T$, such that $\tau \neq \tau'$, it holds that $g(\tau) \cap g(\tau') = \emptyset$.

Given tasks T and agents A, G denotes the family of all feasible team allocation functions. Then, the overall affinity of an FTAF g equals the product over the competence affinity of every team for their assigned task according to g. As we mentioned earlier, the product favours balance, i.e., allocations where every team formed is more or less equally good for their task. The optimum FTAF g^* is the one that maximises the overall team affinity:

Definition 6.4 (Non-Overlapping Many Teams to Many Tasks (NOMTMT) Allocation Problem). Given a set of tasks T, and a set of agents A, the *Non-Overlapping Many Teams to Many Tasks Allocation Problem* is to find the team allocation function $g^* \in G$ that maximises the overall team affinity of the team allocation:

$$g^* = \arg\max_{g\in G} \prod_{\tau\in T} \mathrm{aff}(g(\tau), \tau, \eta^*_{\tau\to g(\tau)}) \tag{5}$$

Here we highlight that for computing the overall team affinity for some FCAF $g \in G$, we need to compute the optimal FCAF for each team for their assigned task according to g. Thus solving the NOMTM allocation problem requires

solving $|T|$ optimisation problems for each $g \in G$..[5] Next, we show that the NOMTMT allocation problem is $\mathcal{NP}$-hard.

Theorem 6.5 *The Non-Overlapping Many Teams to Many Tasks (NOMTMT) allocation problem is $\mathcal{NP}$-hard.*

We omit the proof due to space limitations. However, in Appendix B we show that the problem is $\mathcal{NP}$-hardby using a reduction from *binary combinatorial auction winner determination problem* for single-unit auctions, which is known to be $\mathcal{NP}$-hard [40].

6.3 Solving the team allocation problem: a linear programming approach

We can optimally solve the NOMTMT allocation problem with the means of a linear program (LP); thus, here, we provide the corresponding LP encoding. Let us start, though, with a non-linear encoding. We use a decision variable x_τ^K for each task $\tau \in T$ and each size-compliant team $K \in \mathcal{K}_\tau$. x_τ^K equals 1 when team K is allocated to work on task τ in the optimal solution, and 0 otherwise. Then the non-linear program is:

$$\max \prod_{\tau \in T} \prod_{k \in \mathcal{K}_\tau} \left(\text{aff}(K, \tau, \eta^*_{\tau \to K}) \right)^{x_K^\tau} \tag{6}$$

subject to:

$$\sum_{K \subseteq A} x_K^\tau \cdot \mathbb{1}_{K \in \mathcal{K}_\tau} \leq 1 \qquad \forall \tau \in T \tag{6a}$$

$$\sum_{\tau \in T} \sum_{K \subseteq A} x_K^\tau \cdot \mathbb{1}_{a \in K} \cdot \mathbb{1}_{K \in \mathcal{K}_\tau} \leq 1 \qquad \forall a \in A \tag{6b}$$

$$x_K^\tau \in \{0, 1\} \qquad \forall K \subseteq A, \tau \in T \tag{6c}$$

Then, for the optimal allocation function g^* it holds that $g^*(\tau) = K$ if and only if $x_\tau^K = 1$. Constraints (6b) and (6a) ensures that the resulted g^* is an FTAF (Def 6.3). As we said, the above is not a linear program due to the non-linear objective function (Eq. (6)). However, we can do a linear transformation by considering the logarithm. Thus, we can obtain the optimal allocation function by solving the equivalent linear program below:

$$\max \sum_{\tau \in T} \sum_{K \in \mathcal{K}_\tau} x_K^\tau \cdot \log \left(1 + \text{aff}(K, \tau, \eta^*_{\tau \to K}) \right) \tag{7}$$

subject to: Eqs. (6a), (6b), and (6c). To obtain the equivalent LP, we first shift the function's domain—we do so in order to avoid computing log(0)—and then we use the logarithm to convert the double product into a double sum and power factors into products. *(i)* we use the $\log(\cdot)$ to convert the double product to double sum, and the powered factor into a product; and *(ii)* we change the function's domain to avoid log(0). We can solve this LP with the aid of an off-the-shelf solver (e.g. CPLEX [14], Gurobi [41]), GLPK [42], or SCIP [43]). Given sufficient time, an LP solver will return an optimal solution to the NOMTMT allocation problem.

Notice that to solve the LP above, we need to precompute the competence affinity between each task and every possible size-compliant team, meaning that we need to precompute the optimal CAF (via solving an optimisation problem) for each such ⟨team,task⟩ pair. This is bound to lead to large linear programs as the number of agents and tasks grows.

6.4 Solving the team allocation problem: a heuristic approach

In this Section, we put forward Edu2Com, a novel two-stage heuristic algorithm, to overcome large LPs. In the first stage, we compute an initial feasible solution for the allocation problem, i.e., an initial FTAF (Definition 6.3), while in the second stage, we iteratively improve the initial solution via different *swapping strategies*.

6.4.1 Building an initial team allocation

First, the algorithm builds an initial FTAF. For each task, the algorithm forms a *promising* team, i.e., a team that *seems capable* of tackling the task based on the task's required competencies and the competencies offered by each team member. The teams are formed sequentially, starting with the 'hardest' tasks. We characterise a task as hard when only a few agents can cover its required competencies. With this heuristic, we prioritise the tasks so that 'lighter' tasks (i.e., tasks whose required competencies can be easily covered by many agents) do not bind agents whose acquired competencies can be utilised in hard tasks.

Computing tasks' allocation hardness. The *allocation hardness* (or simply 'hardness') of a task assesses the difficulty of finding agents who can adequately cover the task's required competencies. Intuitively, when, for some competence c, more agents adequately cover c (i.e., with high coverage on c), it is easier to find an agent for some task requiring c, and therefore the task is less hard. Inspired by the notion of *moment of inertia* [44], we measure the difficulty to cover a competence, and therefore the task's hardness requiring that competence, as the effort (distance from (1,0)) that agents should make to reach the ideal competence coverage of c, which is 1. We remind the reader that the coverage of a competence c ranges from 0 to 1. Thus, the ideal com-

[5] Note that for some specific FCAF $g \in G$, computing the $|T|$ optimal FCAFs corresponds to solving $|T|$ *independent* optimisation problems.

petence coverage for a competence occurs if every agent can fully cover the competence (i.e. competence coverage equals 1 for all agents). Let $\mathcal{I}$ be a partition of the competence coverage domain [0, 1] in ten distinct intervals, and $mid(J)$ be the midpoint of the interval $J \in \mathcal{I}$. Now, given a competence $c \in \mathcal{C}$ and a set of agents A, we compute the moment of inertia of c as $I(c) = \sum_{J \in \mathcal{I}} n_J^c \cdot \big(1 - mid(J)\big)^2$, where n_J^c is the number of agents in A whose coverage of competence c lies within interval J, and hence represents the *mass* of c in the interval.

Then the hardness of a task $\tau \in T$ depends on *(i)* the moment of inertia of each required competence $c \in C_\tau$, and *(ii)* each competence's relative importance $w_\tau(c)$. As such, for some task τ we compute τ's hardness as:

$$h(\tau) = \omega \cdot \sum_{c \in C_\tau} w_\tau(c) \cdot I(c) \tag{8}$$

where ω is a normalising factor on the importance weights of all competencies in C_τ.

Building an initial team allocation. When we compute each task's hardness, we build the initial FTAF. To do so, our algorithm, Edu2Com, sequentially for all tasks picks a team out of the available agents, starting from the hardest one. In more detail, let us denote with $A_\tau \subseteq A$ the available agents (i.e., the agents who have not yet been allocated to a team) while picking a team for task τ; and with $\bar{C}_\tau$ task's τ required competences sorted according to their relative importance. The i-th most important competence is denoted as $\bar{C}_\tau^i$. The first agent to be allocated to τ's team is the agent in A_τ that can cover best competence $\bar{C}_\tau^1$—formally, the first agent to be picked is computed as $\sigma_1 = \arg\max_{a \in A_\tau} \{\text{cvg}(\bar{C}_\tau^1, a)\}$. After picking the first agent σ_1, the remaining available agents are $A_\tau - \{\sigma_1\}$. Then, the i-th agent to be picked for the team of task τ is computed as $\sigma_i = \arg\max_{a \in A_\tau - \Sigma_{i-1}} \{\text{cvg}(\bar{C}_\tau^j, a)\}$ where $\Sigma_{i-1} = \bigcup_{k=1}^{i-1} \{\sigma(\bar{C}_\tau^k)\}$ denotes the agents already picked for the team, and $j = \big(i - 1 \pmod{|C_\tau|}\big) + 1$ indicates which competence agent σ_i shall cover best. Thus, the team formed to work on τ is $K = \bigcup_{i=1}^{s_\tau} \sigma_i$, while the agents in K are no longer considered available, i.e., $A_{\tau'} = A_\tau - K$ where τ' is the hardest task after τ.

6.4.2 Improving team allocation

After finding the initial promising FTAF, in the second stage, Edu2Com attempts to improve it iteratively. Specifically, we introduce several heuristics to be applied in each iteration, exploiting swaps of agents among teams. We distinguish two kinds of iterations based on the heuristics applied in each kind of iteration:

1. **Single pairing.** For two randomly selected tasks, Edu2Com applies the following two swapping-based heuristics:
   (a) **Exploiting swap.** Considering solely the agents currently allocated in the selected tasks, find the optimal team allocation.
   (b) **Exploring swap.** If there are available agents (i.e. agents assigned to no team), try to swap a randomly selected *assigned* agent (to any of the two tasks) with a randomly selected *unassigned* agent. Try a maximum of k times, and keep only the swaps that improve the competence affinity.

   Each heuristic is applied (if possible) once within a single paring iteration.
2. **Exhaustive pairing.** For *every* pair of tasks, swap every possible pair of agents within them. If competence affinity improves, keep the change and stop the exhaustive pairing iteration.

The iterative process lasts until *(i)* the global maximum competence affinity is reached, *(ii)* no solution improvement occurs for a number of iterations, or *(iii)* the algorithm is stopped by the user. In all cases, the most recently found solution is returned.

6.5 Handling students' preferences

Notice that although the description of our case study in Section 2 considers students' preferences, the definition of our task allocation problem in Section 6.2 focuses on competence affinity. However, our problem definition can be readily extended to handle preferences. Next, we show how to achieve that by turning the single-objective optimisation problem in Equation 5 into a multi-objective optimisation problem that accommodates both competencies and students' satisfaction. Thus, we propose to follow a similar approach to our own work in [28], where we composed teams considering both competencies and personalities.

Formally, we consider that the preferences for each student a in A can be represented through a linear order (or a ranking) $\succeq_a$ over the tasks in T. We will refer to this order as the preference profile of a. We will note as $\succeq_A$ the set of preference profiles of agents in A. Then, given a task $\tau \in T$, we can compute the satisfaction degree of student a as:

$$sat(a, \succeq_a, \tau) = \frac{|T| - pos(\succeq_a, \tau) + 1}{|T|} \tag{9}$$

where $pos(\succeq_a, \tau)$ stands for the position of τ in the preference profile $\succeq_a$, being 1 the position of the most preferred task, and $|T|$ the position of the least preferred task. Again, since we want to guarantee a balanced preference satisfac-

tion within a team, we compute the satisfaction degree of a team as follows:

$$sat(K, \succeq_A, \tau) = \prod_{a \in K} sat(a, \succeq_a, \tau) \tag{10}$$

Intuitively within a team K, we reach maximum satisfaction when the assigned task τ is the first choice of all members in K, i.e., τ is the most highly ranked task of all for every $a \in K$:

$$\nexists\, \tau' \in T : \tau' \succ_a \tau$$

On the other hand, if τ is least proffered even for a single student, the satisfaction of the team is dropped, and of course, the minimum satisfaction is reached when the assigned task τ is the least ranked task for every $a \in K$:

$$\nexists\, \tau' \in T : \tau \succeq_a \tau'$$

Now we are ready to embed the preference satisfaction of a team in our problem by adding a further objective in the objective function of Equation 6, which then becomes:

$$\max \prod_{\tau \in T} \prod_{K \in \mathcal{K}_\tau} (w_a \cdot \text{aff}(K, \tau, \eta^*_{\tau \to K}) + w_s \cdot sat(K, \succeq_A, \tau))^{x^\tau_K} \tag{11}$$

We use weighting parameters w_a and w_s to regulate the influence of the team's competence affinity and satisfaction, respectively, in the multi-objective optimisation function. Hence we can obtain a family of objective functions including two edge points *(i)* considering the competence affinity ($w_a = 1$ and $w_s = 0$) solely, and *(ii)* considering the team's satisfaction ($w_a = 0$ and $w_s = 1$) solely. It also includes any function that *(iii)* proportionally considers the two parties ($w_a + w_s = 1$ with $w_a > 0$ and $w_s > 0$).

7 Empirical analysis

Our empirical analysis aims to evaluate each of the core AI services that compose the AI4Citizen pilot. Unfortunately, we had to fragment our evaluation and conduct separate empirical analyses on each component due to the pandemic (COVID-19) since the School-Work Alternation programme was cancelled. Thus, Section 7.1 empirically explores using FastText that yields better performance. Thereafter, in Section 7.2, we involve high-school students in the evaluation of the usability of our chatbot. Finally, in Section 7.3, we analyse the time required by our team formation service to provide allocations over real-world data. Furthermore, we conduct a validation of the algorithm's recommended allocations with the aid of experts. Thanks to the results of our empirical analysis, we conclude that the AI4Citizen pilot is ready for deployment in a real-world setting.

7.1 Evaluation of competence and skill extraction

This section aims to evaluate the extraction capabilities of the CSE tool introduced in Section 4. Recall that the CSE tool results from combining the ESCO Ontology [13] and FastText [16], allowing us to relate the texts taken from a CV with job offers by associating them to the competencies in the ESCO Ontology.

To evaluate the CSE tool, we performed two different experiments. In both experiments, we used competencies, CVs, and job offers described in Italian. For the first experiment, we adopted the FastText similarity algorithm out-of-the-box: *(i)* first, the description of the competence in natural language is converted into a vector; *(ii)* similarly, vectors are also computed from all the competences present in ESCO, written in the same language of the required competence; *(iii)* the cosine between the vector in *(i)* and each of the vectors in *(ii)* is calculated, and the results are ranked. The higher the cosine, the more similar the texts behind.

Let d be the text we want to compare with the competencies in ESCO. First, all the documents (job descriptions, CVs, ESCO descriptions of competencies) are pre-processed using Tint [45], a tool for analysing texts written in Italian. With Tint, we perform some pre-processing steps such as tokenisation, sentence splitting, and lemmatization[6], so that words such as *andavano*, *vado* are all referred to the main form *andare*.

After that, vectors for each word are obtained using FastText. A document can therefore be represented as an ordered list of vectors:

$$d = (v_1, v_2, \ldots, v_{n_d})$$

where v_i is the vector representation of the word w_i, being n_d the length of the document in term of words.

Following [46], all the vectors can be merged into a single one by computing the average of them, document by document. Each document d is therefore transformed in a single vector v_d as follows:

$$v_d = \frac{v_1 + v_2 + \cdots + v_{n_d}}{n_d}.$$

[6] There are common tasks in Natural Language Processing: (i) the *tokenisation* splits the text into words; (ii) the *sentence splitting* phase tries to understand, using a mix of heuristics and machine learning, where sentence boundaries are; (iii) the *lemmatisation* provide the base lemma for the words, a task which difficulty depends on the language, and it is particularly hard for Italian.

Finally, we use as a similarity function the cosine between two vectors. Let d and e be two documents, and v_d and v_e be their corresponding vectors. The cosine similarity between them is calculated as follows:

$$\cos\alpha = \frac{\overline{v_d} \cdot \overline{v_e}}{|\overline{v_d}| \cdot |\overline{v_d}|}$$

where α is the angle between v_d and v_e. The smaller the angle (and therefore, the larger the cosine), the more similar are the two documents.

For the second experiment, we added a pre-processing phase where only the most relevant words are taken into account, using TF-IDF (term frequency-inverse document frequency) [47], using an approach similar to the one described in [48].

TF-IDF is a statistical measure that reflects the importance of a word in a set of documents. The intuition is that a term occurring in many documents is not a good discriminator, and hence it should be given less weight than words that occur in fewer documents. The TF-IDF function measures the importance of a term in a particular document: it grows proportionally to the number of times the terms are used in the document, but it decreases proportionally to the frequency of the term in the whole collection.

The function is the product of the two components, TF and IDF. Let t be the term and d the document:

$$\mathrm{tf}(t, d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}$$

$$\mathrm{idf}(t, D) = \log \frac{N}{1 + |\{d \in D : t \in d\}|}$$

where $N = |D|$ is the number of documents in the set D.

Once we have computed the TD-IDF value for each term and document, our goal is to use that function to remove from the set of words both the terms that are too rare and those that—conversely—are too common in the ESCO corpus. We then set two parameters, $f_{\min}$ and $w_{\min}$, respectively, the minimum frequency of a word in the ESCO ontology and the minimum TF-IDF weight. We test different settings with a matrix of possible values and find out that the best values *wrt.* our evaluation is $f_{\min} = 5$ and $w_{\min} = 2.0$. We then filter out terms that appear less than 5 times in the whole set of documents and words whose TF-IDF value is less than 2.0.

After these pre-processing steps, documents are processed as already described above.

To evaluate the approaches, we ask teachers from high schools in Italy to annotate some sentences written in Italian taken from the "Alternanza Scuola-Lavoro" project, plus a set of custom sentences written from scratch by the teachers themselves. The first ten competencies from ESCO were shown for each sentence, and teachers were asked to annotate whether they were relevant to the input sentence or not. A total of 20 teachers took part in the experiment, annotating 368 pre-defined sentences and 65 manually written sentences. Table 2 shows the number of sentences that were annotated as correct and incorrect in the first two ESCO competencies of the ranking.

Table 2 Accuracies extracting competencies from free text

Sentences	TF-IDF + FastText		FastText	
	correct	incorrect	correct	incorrect
Predefined	309	127	287	145
Manually written	53	56	47	61

Results show that both algorithms are effective on predefined sentences, and in particular, the pre-processing using TF-IDF increases the accuracy compared to the plain use of FastText. By analysing the sentences the teachers wrote, we can see a massive difference in the ratio between correct and incorrect in the two sets. This is due to the freedom left to the teachers, which led them to enter texts that are too long or too short concerning the average descriptions of competencies in ESCO, yielding more challenging examples to the algorithm.

Generally, the system performs well, and our experiments allowed us to understand which algorithm performs better. Most of the inaccuracies are due to the imbalance of the ESCO ontology: some branches are very detailed, while others are often poor or missing at all. Both the approaches (with and without TF-IDF) are easy to extend to any language, as long as both FastText and ESCO are available in that language.

Since the pre-processing phase using TF-IDF outperforms Fastext alone, we use that algorithm in the AI4Citizen pilot. Notice that our algorithm is publicly available on Github [18].

7.2 Evaluation of the chatbot usability

We evaluated the usability of the chatbot involving three high school classes of two different technical institutes for a total of 55 students:

- 20 students from a fourth class of a technical institute (referred to as Class1)
- 17 students from a fifth class of the same institute (referred to as Class2)
- 18 students of a fourth class of another institute (referred to as Class3)

We only involved students of technical institutes because these types of schools require their students to do long periods of SWA (school-work alternation) activities.

Table 3 CUQ scores per class

	CUQ Score	Lowest Score	Highest Score	Median Score
Class1	41.1 ± 11.5	26.56	68.75	39.06
Class2	58.0 ± 21.9	21.90	85.94	64.10
Class3	63.0 ± 13.8	37.50	89.10	61.70

7.2.1 Procedure

The students of Class1 and Class2 were in their classroom assisted by their professor, whereas the researchers who organised the evaluation were remotely connected to the classroom due to the constraints imposed by the pandemic.

The students of Class3 were remotely connected to a virtual meeting room. Two professors chaired the evaluation activity from the school classroom connected to the virtual meeting room.

The evaluation activity consisted of three phases:

1. Each student logged into the chatbot using their private institute's credentials and worked on a given mission.
2. After completing the mission, they filled in a questionnaire composed of the Chatbot Usability Questionnaire (CUQ) plus four open questions (see Appendix A for detail about the questionnaire).
3. Finally, the researchers give a short presentation of the AI4EU project to inform students about the context of the evaluation test.

The mission given to the students consisted of *(a)* understanding if they had to do an internship this school year and *(b)* choosing three internships of interest. Class3 also had a third mission consisting of *(c)* looking at how to reach the companies of the formerly chosen internships and opening their website.

We enhanced the chatbot between the different assessment phases within this real-life experimentation. Since the students interacted freely with the chatbot, we enriched the intent recognition with the students' expressions using the platform's monitoring feature. We also identified that a few students were too restrictive in expressing their preferences, which ultimately did not enable them to complete their mission (*e.g.,* the selection of 3 choices), causing frustrations. We modified our chatbot to guide them better to alleviate these restrictions and select other internships. We were also enabled to connect to the company website and to look at their precise locations taking into account informal remarks from the first class. The improvement of the results (see Table 3) showed how these relatively simple enhancements drastically changed the student experience and enabled a more significant proportion to complete their mission (see Table 4).

7.2.2 Results

The scores of the CUQ questionnaire filled in by the students are reported in Table 3. When asked if it was easy to solve the given mission, 65% of the participants (36 students) said yes (see Table 4 for details). When asked what they liked most about the chatbot, 21 students (38% of the participants) answered that they liked to receive information on internship offers; 16 students (30%) answered they liked the interaction with the chatbot. When asked about suggestions for improving the chatbot, we got many interesting answers that helped improve it. Seventeen students recommended expanding the vocabulary of allowed inputs. They would have preferred more tolerance in the type of questions they could enter.

7.2.3 Discussion of the results

From the data obtained with the CUQ questionnaire, it emerges that usability problems still need to be solved, even if the score has improved from the first to the third experimentation (see Table 4).

Despite the fact that the chatbot still has usability issues, students liked being able to use this modality to get information about their internship.

Interestingly, our results showed that students still believe that interacting with a chatbot is the same as interacting with a person. They would like to use a completely free vocabulary, not constrained at all. This issue has to be considered in future versions of the chatbot.

7.3 Evaluating the team allocation algorithm

In this section, we evaluate the capability of our team allocation algorithm to cope with real-world problems. Moreover, we evaluate the quality of the team allocations that it computes with the aid of experts. More precisely, Section 7.3.2 details the analysis of the solution quality, runtime perfor-

Table 4 Answers for question Q17

Q17. Was it easy to solve the task?				
Answer	Class1	Class2	Class3	Total
Yes	11	10	15	36
No	9	7	3	19

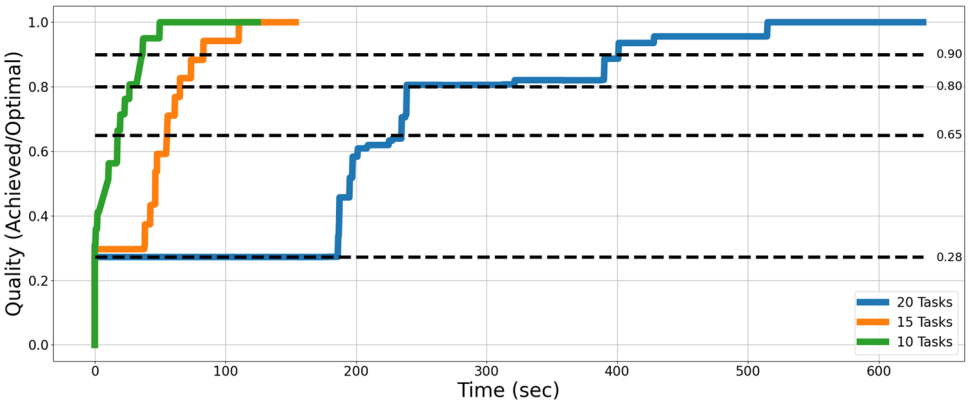


Fig. 7 Competence quality of our algorithm along time

mance and anytime capabilities of Edu2Com when pitched against CPLEX V12.10.0, a state-of-the-art linear programming solver, over synthetic data. Section 7.3.3 reports on the solution quality and time performance of Edu2Com when solving several problem instances with real-world data. Finally, Section 7.3.4 details the validation of the quality of allocations by experts.

7.3.1 Empirical settings

The implementation of Edu2Com, along with all the necessary supporting code, was made in Python3.7. All the experiments ran on a PC with Intel Core i7 CPU, 8 cores, and 8Gb RAM. In our experiments below, we set our algorithm's parameters to: compute similarity with $\kappa = 0.35, \lambda = 0.75$; perform one exhaustive-pairing every 50 single-pairings; stop the algorithm after two blocks of single-pairings and exhaustive pairings have elapsed with no improvements.

7.3.2 Quality, runtime and anytime analysis

Synthetic Data Generation. To compare our heuristic algorithm against the optimal solver, we created three sets of problem instances with varying sizes (small, medium, and large), all with solutions that CPLEX could obtain within acceptable time limits. The problem instances were created with the process described below. Initially, we determined the number of tasks by selecting from the set {10, 15, 20}. Then, for each task τ, we randomly selected: the team size $s_\tau \sim \mathcal{U}(2, 3)$, the number of required competencies $|C_\tau| \sim \mathcal{U}(2, 5)$, the set of $|C_\tau|$ competencies from the ESCO ontology, and an importance weight for each competence $c \in C_\tau$ as $w_\tau(c) \sim \mathcal{N}\big(\mu = \mathcal{U}(0, 1), \sigma \sim \mathcal{U}(0.01, 0.1)\big)$.[7] After that, for each task τ, we created s_τ agents so that each agent acquires competencies that either a required competence in C_τ or a child node (in ESCO) of a required competence. In total, we evaluated solving 60 problem instances of varying sizes. Specifically, we considered three data families containing 20 instances each, where each problem instance involves (1) 10 tasks and on average $\sim$ 24.5 agents in the low-size family, (2) 15 tasks and on average $\sim$ 37.7 agents in the middle-size family, and (3) 20 tasks and on average $\sim$ 50.55 agents in the large-size family.

Quality analysis. Figure 7 illustrates the *quality* of the solutions build with our algorithm, Edu2Com, across time. As quality, we consider the ratio of the overall competence affinity of a solution against the competence affinity of the optimal solution computed by CPLEX. In the figure, we see the average of over 20 problem instances per data family (low-size, middle-size, large-size). As we can see, Edu2Com always reaches the optimal solution. That is, our algorithm exhibits quality 1 in all instances across every data family.

Runtime analysis. The greatest advantage of our heuristic algorithm is that it is much faster than CPLEX. Table 5 shows the time we can save with respect to CPLEX to reach optimality. Overall, using our heuristic can save from ~52% to ~65% time with respect to CPLEX. Specifically, for problem instances with 10 tasks (small scenario), we save $\sim$ 56% time *wrt.* CPLEX; for problem instances with 15 tasks (medium scenario), we save $\sim$ 52% time; and, for problem instances with 20 tasks (large scenario), we save $\sim$ 65% time *wrt.* CPLEX. We should note that the primary time-consuming task for CPLEX is building the LP encoding the problem.

Anytime analysis. Our algorithm reaches high-quality solutions (with quality above 0.8) in $\sim$ 6.4% to $\sim$ 12.6% of the time required by CPLEX to yield a solution. Let t_{opt} be the time in seconds that CPLEX needs to yield a solution.

Table 5 Time savings to converge to an optimal solution *wrt.* CPLEX

Scenario	Time Savings *wrt.* CPLEX(%)
Small-Size (10 Tasks)	56%
Medium-Size (15 Tasks)	52%
Large-Size (20 Tasks)	65%

[7] Note that any weight sampled below 0 is considered as 0, and any weight above 1 is considered as 1.

Table 6 Quality of the solution as time progresses, the time needed in seconds, and the proportion of time compared to the time required by CPLEX (t_{opt})

Quality	Small Scenario (10 Tasks)		Medium Scenario (15 Tasks)		Large Scenario (20 Tasks)	
	Time (sec)	Portion of t_{opt}	Time (sec)	Portion of t_{opt}	Time (sec)	Portion of t_{opt}
≥30%	0.091	$2.25 \cdot 10^{-5}$	1.451	$3.31 \cdot 10^{-3}$	2.564	$1.135 \cdot 10^{-3}$
≥65%	17.51	0.043	54.608	0.125	227.93	0.119
≥80%	26.313	0.064	65.251	0.149	238.715	0.126
≥90%	36.061	0.088	83.205	0.189	400.796	0.219
100%	49.806	0.122	110.38	0.252	515.085	0.271

Edu2Com finds the first solution: (1) after $2.25 \cdot 10^{-5} \cdot t_{opt}$ (=0.091) seconds with a quality 32% of the quality of the optimal in the small-size scenario, (2) after $3.31 \cdot 10^{-3} \cdot t_{opt}$ (=1.451) seconds and 30% in the medium-size scenario, and (3) after $1.35 \cdot 10^{-3} \cdot t_{opt}$ (=2.564) seconds and 28% in the large-size scenario. In Table 6, we provide the time needed by Edu2Com to reach a solution of a certain quality and the portion of time compared to t_{opt}. Notably, our algorithm reaches a solution of quality above 80% in (1) $0.064 \cdot t_{opt}$ seconds in the small-size scenario, (2) $0.149 \cdot t_{opt}$ in the medium-size scenario, and (3) $0.126 \cdot t_{opt}$ seconds in the large-size scenario.

7.3.3 Solving real-world problems

In this empirical analysis section, we used *real-world data*. In more detail, we use 100 profiles of students and 50 internship programs, with the competencies described in the ESCO ontology [13]. There are 118 distinct competencies in the students' profiles, each acquiring $\sim$ 11.98 competencies, while there are 34 distinct competencies in the internship programs' description, each requiring 4 competencies (minimum 2, maximum 15).

This analysis aims to highlight the scalability of the problem as tasks' required team sizes increase and assess Edu2Com on handling the problem. Notably, the actual data regarding the tasks (internship programs) did not specify the team size. Thus we synthetically populated the problem instances with specific team sizes. [28] points out that teams within classrooms shall have at most five members. As such, we considered problem instances where all tasks require the same team size, ranging in [2, 5]. Moreover, we considered problem instances with varying required team sizes. Specifically, the problem instances contained tasks requiring team sizes ranging in [2,3], [2,4], [2,5], [3,4], [3,5], and [4,5]. In each problem instance, the number of tasks requiring a specific team size is uniformly distributed across the team sizes in the corresponding range. As we discuss in Section 7.3.5, we cannot optimally solve instances with 100 agents, 50 tasks, and the aforementioned team sizes.

Analysis. In Fig. 8, we show our findings with respect to the time needed by our algorithm to converge to a solution. Each bar in Fig. 8 illustrates the average time (in minutes: seconds) over 20 problem instances per required team size. Here we highlight that Edu2Com converge to a solution in less than 50 minutes, especially in large settings that contain tasks requiring teams of size 5. Our experiments showed that Edu2Com needs less time to solve instances containing tasks requiring smaller team sizes—such a result is expected since the search space is much smaller in small team-size instances. Specifically, we observe that solving a problem instance with team size 2 requires less time than a problem instance with team size 3, which in its turn requires less time than a problem instance with team size 4, which in its turn requires less time than a problem instance with team size 5. Moreover, notice the time needed for finding a solution in problem instances with tasks requiring team sizes in a range $[a, b]$. In such problem instances, the time needed to find a solution falls between the time Edu2Com needs to solve *(a)* problem instances requiring teams only of size either a or b, and *(b)* instances requiring teams in team sizes in ranges $[a, b-1]$ and $[a+1, b]$. In the hardest scenario where all tasks' required team size is size 5, Edu2Com yields a solution in (approximately) less than 50 minutes. Given that this process does not need to be performed in real-time with highly demanding time constraints (i.e., we do not need to come up with a solution within a few seconds), it is acceptable.

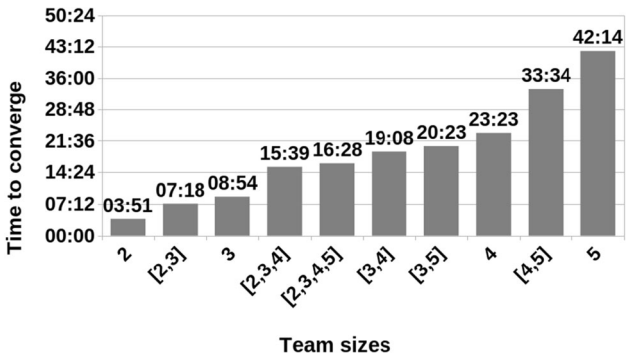


Fig. 8 Time in Min:Sec required as team sizes grow

Notably, the educational authorities spend much more working time matching students to internships since the current practice is to do it manually; while solving the problem optimally with the means of an LP is infeasible since we cannot even generate the LP in time. Hence, our findings confirm our algorithm's feasibility in large problem instances and show that Edu2Com can handle the team allocation problem in this education scenario.

7.3.4 Validation by experts

In the last part of our empirical analysis, we focus on validating our algorithm by educational authorities with experience in allocating students to internship programs. To do so, we pitched Edu2Com against teachers with such experience—we refer to them as *experts*. In more detail, we consider a synthetic problem instance—similar to the actual-world problem instances we employed in Section 7.3.3—with 100 agents (students) and 50 tasks (internships) requiring teams of sizes 1, 2 and 3. The problem instance used here is the largest regarding the number of tasks we can generate with the 100 student profiles at hand. Notably, solving this problem optimally (*e.g.,* with CPLEX) would require more than 1.8 million decision variables.

Then, we followed the process below. Given the synthetically generated problem instance: *(1)* an expert matches each internship with a student team by hand, *(2)* Edu2Com computes an allocation of students to internships, and *(3)* we randomly allocate a student team to each internship. Henceforth, g_{expert}, g_{edu2com}, and g_{random} stand for the allocations built with each method, respectively. Next, we task eight experts with experience in the allocating process to assess and compare the three allocations, ignoring by which method each allocated was built—we refer to these experts as *evaluators*. Notably, regarding the time needed to reach a solution, Edu2Com required less than 1 hour and 45 minutes to build an allocation. In contrast, the expert needed approximately the time of a working week, including studying, analysing and matching each internship with a student team.

Evaluation Process. We asked each evaluator to study and assess the three allocations. Specifically, the evaluators should mark the team allocated to each internship as follows: 1 for high quality, 2 for medium quality, and 3 for low quality. Moreover, the evaluators were allowed to mark with the same value teams assigned to the same internship that were produced by different methods in case they considered the teams were of the same quality

Handling missing data During the final analysis, we encountered a challenge of missing data where the expert could not form a team for every internship. Specifically, out of the 50 internships, the expert failed to provide a team for 13 of them, resulting in 23 students out of 100 being without internships. As a result, the evaluators had to work with incomplete data and provide incomplete evaluations. The absence of expert allocation for some tasks prevented the evaluators from rating all three allocation methods (g_{expert}, g_{edu2com}, and g_{random}). An auxiliary mark of 4 was used to signify the missing allocation, indicating absence, which was considered worse than a low-quality mark (mark 3). As a result, all evaluators marked any missing allocation with a four. Additionally, the evaluators missed marking the teams of some internships, resulting in low-quality marks (mark 3) for missing evaluations.

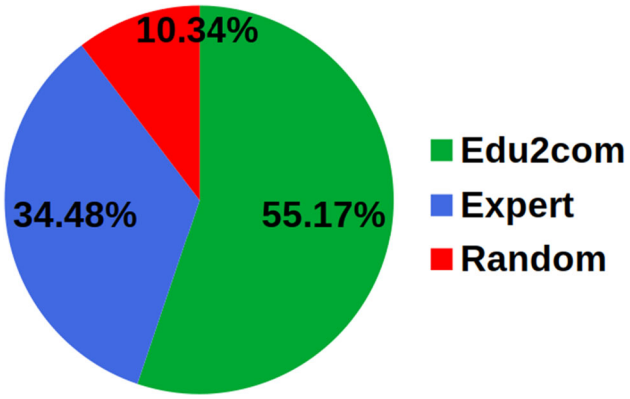


Fig. 9 **Single winner**: Edu2Com vs Expert vs Random. Percentage out of 29 tournaments

Analysis. Our analysis aims to determine the best method for allocating student teams to internships, founded on the evaluators' assessments. For that, we consider the evaluation of the teams assigned to each internship as a *tournament*, while a tournament involves three rounds of competing pairs: (1) Edu2Com vs Expert; (2) Edu2Com vs Random; and (3) Expert vs Random.

The evaluators' marks serve as the means for determining the winning allocation method for each round and, therefore, for each tournament. The allocation method that receives the higher aggregated mark in a round is declared the winner of that round and earns one point for the corresponding internship assignment. If in a round two internship assignments are

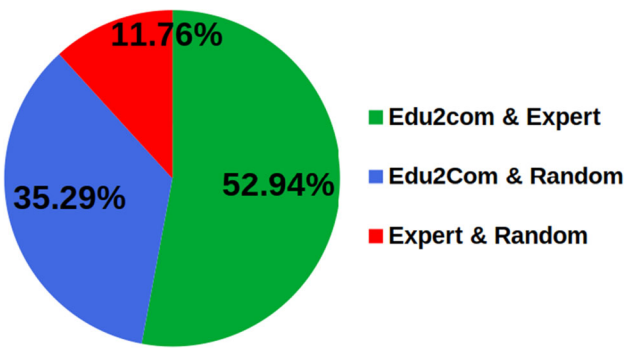
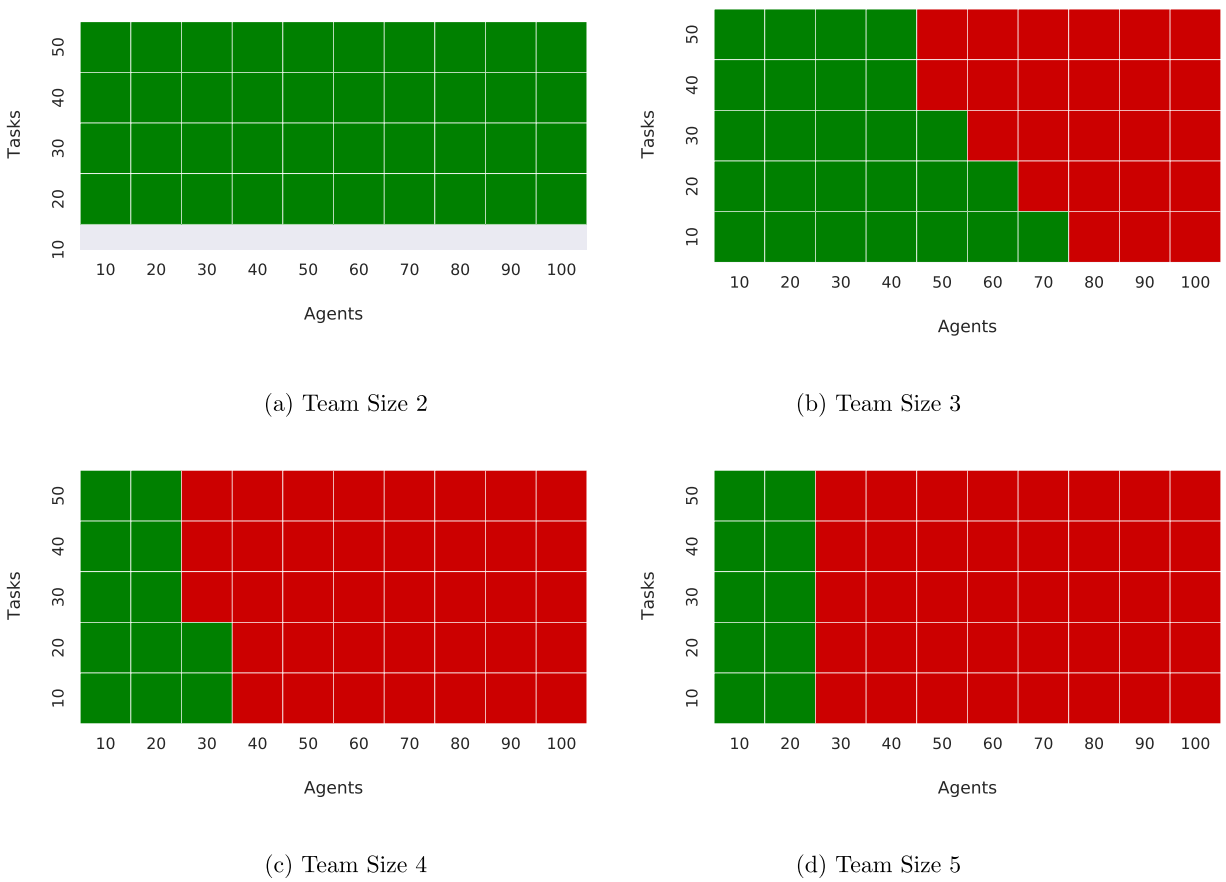


Fig. 10 **Tie with two winners**: Edu2Com vs Expert vs Random. Percentage out of 17 tournaments

of the same quality, i.e. there is a tie, both of their allocation methods receive half a point. By tallying the points earned from each round of the tournament, we utilise the Copeland$_\alpha$ voting rule [49] (with $\alpha = 0.5$) to determine the overall winner. This voting rule is shown in [50] to be "resistant to all the standard types of (constructive) electoral control." In a nutshell, the allocation method that accumulates the highest number of points throughout the three rounds is declared the tournament winner. Again, if there is a tie between two allocation methods, each one earns half a point. For instance, suppose that for a given tournament, 8 evaluators considered the heuristic algorithm (Edu2Com) the best assignment, 5 evaluators preferred the expert's allocation to the random one, and 2 evaluators equally favoured both the expert and the random allocation. This would result in the heuristic algorithm earning 8 points, the expert's allocation receiving 6 points, and the random allocation getting 2 points. Therefore, the tournament winner, in this case, would be the heuristic algorithm.

In each tournament, there can be three outcomes: *(i)* a single winner, *(ii)* a tie with two winners, or *(ii)* no winner. Our analysis shows that 58% out of 50 tournaments declared a single winner, 34% declared two winners in a tie, and 8% declared no winner. Figure 9 illustrates the tournament results that declared a *single winner*. As we can see, 55.17% of these tournaments were announced as the winning allocation method Edu2com, the heuristic algorithm, 34.48% of them were announced as the winning allocation method the human exert, while 10.35% of the tournaments were won by the random allocation method. As such, our heuristic algorithm, Edu2Com, was the preferred allocation method to allocate student teams to internships.

Now, in Fig. 10, we report the tournaments announcing *a tie with two winners*. As expected, the majority of the tournaments (52.92%) declared a tie between the methods of our heuristic algorithm and the human expert. Overall, Edu2Com was announced as a winner in a tie in 88.23% of the tournaments. To summarise, our analysis indicates



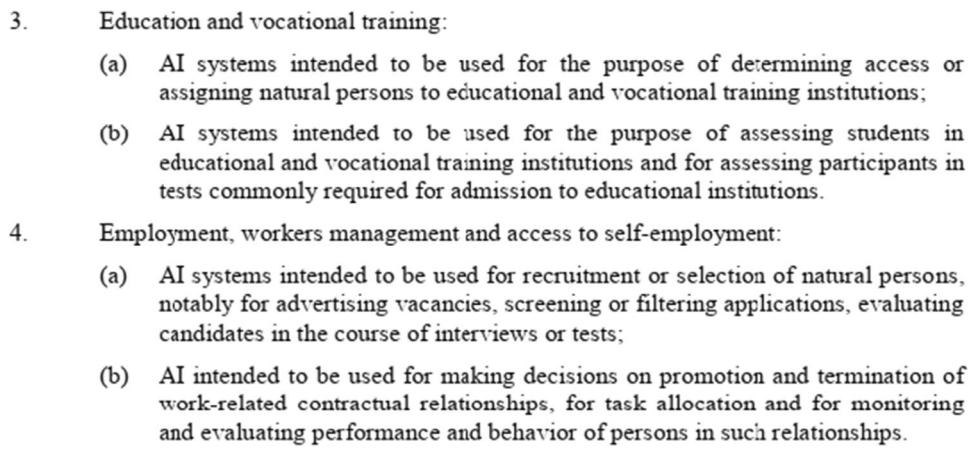
(a) Team Size 2

(b) Team Size 3

(c) Team Size 4

(d) Team Size 5

Fig. 11 Problem instance configurations (number of agents, number of tasks, and team sizes) solvable by CPLEX. Red squares correspond to configurations that CPLEX cannot solve (because it runs out of memory), while green squares correspond to configurations that CPLEX can solve



3. Education and vocational training:
   (a) AI systems intended to be used for the purpose of determining access or assigning natural persons to educational and vocational training institutions;
   (b) AI systems intended to be used for the purpose of assessing students in educational and vocational training institutions and for assessing participants in tests commonly required for admission to educational institutions.
4. Employment, workers management and access to self-employment:
   (a) AI systems intended to be used for recruitment or selection of natural persons, notably for advertising vacancies, screening or filtering applications, evaluating candidates in the course of interviews or tests;
   (b) AI intended to be used for making decisions on promotion and termination of work-related contractual relationships, for task allocation and for monitoring and evaluating performance and behavior of persons in such relationships.

Fig. 12 High-risk AI scenarios described in Annex III of European AI Act proposal

that expert evaluators deem our proposed heuristic algorithm as the method of choice to assign teams of students to internships.

7.3.5 Reasonable time limits for response and limitations of optimal solving

In Sections 7.3.2-7.3.4, we systematically evaluated our proposed heuristic algorithm, Edu2Com. First, we pitched Edu2Com against CPLEX, and then we challenged it to solve real-world problem instances. Notably, the problem we address here does not require a real-time solution within tight time limits. As we stated in Section 7.3.4, a human expert may require a whole week in order to arrange a real-world school-work alternation programme, namely to solve a real-world problem instance with around 100 students and 50 internships. As such, we believe that 24 hours is a reasonable computing time for solving the problem. As mentioned in Section 7.3.3, CPLEX cannot cope with real-world instances. The purpose of this section is to empirically characterise the instances that we can optimally solve, and hence the limits of CPLEX as requested by the reviewer. Figure 11 shows the configurations of problem instances (in terms of the number of agents, number of tasks, and team sizes) that CPLEX can optimally solve. Green squares show problem instance configurations that CPLEX managed to solve, whereas red squares show problem instance configurations that CPLEX could not solve (because it runs out of memory). The figure shows that CPLEX cannot handle real-world problem instances (with 100 agents and 50 internships) unless we limit team size to 2. This is very limiting because it forces us to team up agents in pairs, which is not realistic because, as pointed out in [28], educational scenarios require teams of sizes up to 5. As we increase the team size from 2 to 5, the range of configuration of problem instances that CPLEX can optimally solve dramatically decreases. We empirically observe that CPLEX did not manage to solve problem instances that lead to an LP containing more than around $8 \cdot 10^5$ decision variables. Figure 11 characterises the *frontier* of configurations of problem instances that CPLEX can optimally solve.[8] We highlight that the solving time depends on *(1)* the number of agents, *(2)* the number of tasks, and *(3)* the required team sizes (by all tasks), with the number of agents and team size being the most influential.

8 Privacy and responsible AI

From the inception of the AI4EU project, the AI4Citizen pilot had a special duty to address privacy and responsible AI thoughtfully. These topics are critical for accepting AI-based services run by governments and government agencies and reducing the fear of dystopian scenarios. The SWA case study itself is particularly demanding since the system needs to interact with minors, which are considered vulnerable by the GDPR [51]. Supporting the decision of the recommendation for internships is also critical and, albeit different, closely related to the high-risk scenarios identified in the recent proposed European Artificial Intelligence Act [52] shown in Fig. 12 concerning vocational training and employment.

During the design phase, considering the GDPR and in an attempt to comply with the existing contract between FBK and the region of Trento, we designed our solution to minimise the personal information exchanged between the system components. For example, the match between students and internships is done by a component hosted within the FBK network. Similarly, the matching algorithm only manipulates a set of preferences without access to students' names or other PII. The information exchanged by the components is clearly described by a set of public APIs avoiding direct access to the underlying database. Finally, commu-

[8] In the Appendix C we include the *frontier* of configurations of problem instances along with the number of decision variables involved in the corresponding LPs and required solving time.

nication is done over secure communication channels with client authentication.

For the usage of AI, we analysed the intent of the automation and the impact on the two types of users - students and professors - of the solution. For students, our solution gathers the students' preferences by recommending a set of proposals based on the tagging done by the school and automated mapping to the job ontology, the fit between the individuals' offered skills and the internship required skills (provided as information). At the same time, nothing precludes a student from choosing an internship that exhibits a lousy fit. Since we lack feedback information (see Section 9), we, unfortunately, cannot currently provide further recommendations based on former students' experiences. The professor ultimately does the practical assignment of students to internships. Our system—through the Edu2Com component— provides a recommendation to rely on based on the preferences expressed by the different students and their acquired skills, mimicking the professor's behaviour but exploring the different combinations exhaustively. Still, to support the professor in taking an informed decision, we provide metrics that capture the overall solution fit, the group fit and the individual fit (see Section 6 for details). In the last version of the system, we also introduced the possibility for the professor to challenge certain choices by asking targeted questions.

To cross-check the soundness of this approach, we evaluated our solution - each component and the global solution - against the preliminary questionnaire provided by the AI4EU ethical experts mainly inspired by the ALTAI guidelines [53]. We did not detect violations of these principles, and all components were below the high-risk level threshold. This collaboration contributed to the evolution of the assessment questionnaire later published in [54].

9 Perspectives for an actual-world deployment

In order to better understand the potential business value of the AI4Citizen scenario, we decided to use the Data-Driven Innovation Framework (DDI) [55] to help us develop a consistent strategy and explore all the dimensions of our use case. The DDI Canvas was developed as a guide in exploring all relevant dimensions on the supply and demand side of data-driven innovation in a systematic manner. Using such a methodology in a research project is unusual but was vital to understand better *(1)* the data required to realise a full fledge version of our proof of concept, *(2)* to identify the network of actors required to achieve this vision and *(3)* to discover some avenues to make such software sustainable. As seen in Fig. 13, we identified the primary proposal value for three personas and the data sources to implement our scenario best.

Concerning data, the main gap was the lack of students' feedback on former internships to enhance the advice provided by the system. Similarly, collecting more data was needed in order to improve the process itself and be able to capitalise on past students' interviews. We strongly recommend that the institutes systematically collect such information and eventually share this data (upon anonymisation) across institutes; this will be a unique way to propose better support to students.

Concerning the ecosystem, while the institutes and the Ministry of Education are the main actors, we believe that the students may benefit from opening up to the business social network, such as LinkedIn, Twitter, and Glassdoor, reflecting the current practices in the business world in alignment with the rationales behind the SWA. While this can only apply to 18 years old students, this will enable them to start building a professional presence and business network, sharing

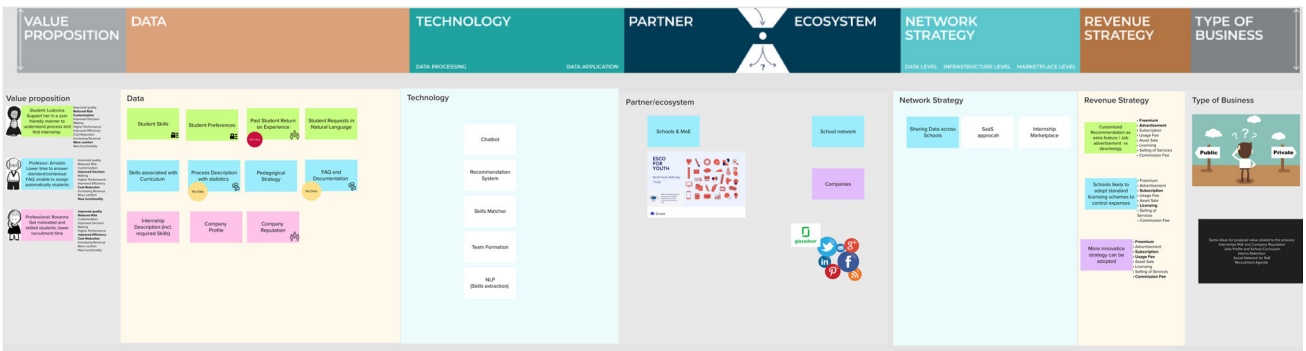


Fig. 13 Business perspective of the SWA scenario using DDI

insights on their experience and eventually getting acknowledgements from their mentors or people from companies. With the proper connectors, this information can also be a way to generate information that can also be retrofitted to the system to enrich a reputation system.

Besides education, we have found further applications of Edu2com, our novel algorithm, to group students into teams. Thus, we have adapted Edu2Com to support team formation in open innovation and team formation for social impact tasks. On the one hand, we have employed Edu2Com to form teams to tackle *innovation challenges* (posed as tasks) in the realm of the H2020 Crowd4SDG project [56] (supporting the events of GEAR Cycle 2 and GEAR Cycle 3). On the other hand, in collaboration with YOMA-Africa [57], we have employed Edu2Com to form teams of young volunteers to carry out social impact tasks. More precisely, we have used Edu2Com to form teams that undertake community clean-up missions[9], matching each team with a location to collectively clean.

Finally, the cost to deploy, maintain and enrich such systems may benefit from a modern approach that leverages both the institutions and other actors. While one can argue that it is against educative system ethics, a too-conservative approach and the impact on resources to develop intelligent software may pave the way for private companies that will propose orientation services that will be accessible only to privileged students.

10 Conclusions

This paper focuses on providing AI-based decision support tools for the School-Work Alternation programme. Our technology, the AI4Citizen pilot, is meant to facilitate the programme's implementation, which is called to support the development of skills for the employability of new generations. Our technology is meant to serve two primary purposes. First, we expect a dramatic reduction in the time required to allocate students to internships. This is because, nowadays, implementing the SWA programme requires much manual intervention from public servants in schools. Second, our technology has been conceived to foster innovation in the programme. This is because our AI-based system allows teams to be allocated to internships, enabling team-based learning and the acquisition of teamwork skills. Since one of the aims of the SWA programme is to spur the development of students' soft skills [2], working in teams would foster the learning of the soft skills that are considered as crucial for collaboration and teamwork [11].

From an applied AI perspective, the AI4Citizen pilot results from *pipelining* a variety of AI technologies: (1) NLP algorithms for extracting competencies and skills *offered* by students in their curricula and *requested* by companies in their internship offers, and for, later on, matching them; (2) a chatbot to assist students in getting acquainted with the internships offered and select the ones they like best; (3) a novel algorithm to group students into teams to allocate them to the internships on offer. Interestingly, the development of each one of the AI technologies in the AI4Citizen pilot results from a different approach. First, the development of our NLP algorithms is founded on state-of-the-art NLP techniques. Second, the development of our chatbot employs commercial chatbot technology. Third, our algorithm to match teams to internships called for novel research contributions. Finally, we conducted an exhaustive empirical analysis of each one of the AI components in the AI4Citizen pilot. Our results indicate that the AI4Citizen pilot is ready for deployment in a real-world setting.

We envision several paths to future research. First, since our results indicate that our pilot is ready for deployment, FBK will investigate plans to gradually integrate its components into their current SWA application, starting with the Competencies and Skills extraction component.

Second, we would like to enrich our experiments along two directions that were planned at the outset of the AI4EU project but were not possible because of COVID-19. On the one hand, along the lines of [10], we would like to compare the performance *at work* of the teams composed by our algorithm versus teams composed by experts to quantify the learning benefits of our approach. On the other hand, it would be interesting to explore the employment of a *coaching* chatbot that will engage students proactively in an event-driven way (*e.g.,* an internship of interest became available, a deadline is approaching, an achievement improved the student's profile) also tackling the following phases of SWA (*e.g.,* internships, feedback collection).

Third, we would like to explore and consider objectives beyond competencies and preference satisfaction when matching student teams to internship programs. In particular, we may consider objectives that the literature in Psychology and Social Sciences has identified to benefit teamwork. Along this direction, an interesting, additional objective to consider is the *diversity of personalities* within teams. As shown in [5, 58], teams with diverse personalities are more effective and perform better (as a team). Another important objective positively affecting teamwork is *social cohesion* [59, 60]. Social cohesion refers to the cohesion of relations between team members. High social cohesion within a team influences the team's performance positively. Similarly, we could consider maximising the consensus among team members or minimising conflicts and disagreements [61]. Furthermore, many educational institutions and

[9] See the public call published by Yoma Africa on Twitter at http://bitly.ws/DUPp.

organisations try to form *diverse* teams concerning nationality or cultural backgrounds to promote cultural exchange.

Finally, a very interesting line of research is that of building contrastive explanations [62] that help explain the team formation algorithm's decisions to students and those in charge of running the programme.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature. The research leading to these results received funding from project AI4EU (H2020-825619). Athina Georgara has received funding from Generalitat de Catalunya under Grant 2019DI17; and has received research support from company Enzyme Advising Group. Furthermore, partial financial support was received from projects: CI-SUSTAIN (PID2019-104156GB-I00); Crowd4SDG (H2020-872944); TAILOR (H2020-952215); VAE (TED2021-131295B-C31), funded by MCIN/AEI/10.13039/501100011033 and by the "European Union NextGenerationEU/PRTR; VALAWAI (Horizon Europe #101070930); and 2021 SGR 00754.

Data availability The data used in the experiment Section 7.3.2 is available from the corresponding author upon request. The students' data we use in the rest of the experiments cannot be disclosed due to GDPR issues.

Appendices

Appendix A. Our questionnaire

In this appendix, we describe the questionnaire we used for evaluating the usability of our chatbot. This questionnaire consisted of the Chatbot Usability Questionnaire (CUQ) plus four open questions. CUQ is a questionnaire specifically designed for measuring the usability of chatbots by an interdisciplinary team from the Ulster University [63]. Chatbots are conversation-driven systems. The most popular usability metrics, such as the System Usability Scale (SUS) [64], may not be suitable for evaluating their usability. However, the CUQ is designed to be comparable to SUS. They are both based on the rating scale proposed in Table 7.

The sixteen questions in the CUQ are shown in Fig. 14.

The four open questions we entered in our questionnaire are the following:

Q17. Was it easy to solve the task?
Q18. What did you like about the chatbot?
Q19. What did you not like about the chatbot?
Q20. Do you have any suggestions for improving the chatbot?

Appendix B. Theorems and proofs

Theorem 6.5 *The Non-Overlapping Many Teams to Many Tasks (NOMTMT) allocation problem is $\mathcal{NP}$-hard.*

Proof We can decide whether a given solution is feasible in polynomial time ($\mathcal{O}(\nu)$ where $\nu = \sum_{\tau \in T} s_\tau$). We now show that the problem is $\mathcal{NP}$-hardby using a reduction from the *binary combinatorial auction winner determination problem (BCAWDP)* for single-unit auctions, which is shown to be $\mathcal{NP}$-hard [40]. An instance of BCAWDP for single-unit auctions is given by a set of items $Items = \{1, 2, \cdots, n\}$, and a set of bids $Bids = \{\mathbf{B}_1, \mathbf{B}_2, \cdots, \mathbf{B}_x\}$. A single bid $\mathbf{B}_j = (S_j, p_j)$ consists of a subset of items $S_j \subseteq Items$, and a price $p_j \geq 0$ that the buyer is willing to pay in order to get the items in S_j. Given an instance of BCAWDP, we construct an instance of the NOMTMT allocation problem as follows: "For each item in $Items$ we create a dummy agent. For each set of bids over κ items, we define the set $\mathcal{B}_\kappa = \{\mathbf{B}_j \in Bids : |S_j| = \kappa\}$. From $\mathcal{B}_\kappa$, we create $|\mathcal{B}_\kappa|$ tasks of size κ. Moreover, given a bid $\mathbf{B}_j \in \mathcal{B}_\kappa$, its items S_j make a candidate team for all tasks of size κ with matching quality p_j. In case there are more than one (e.g., $\xi > 1$) bids over the very same items offering different prices, since $\xi \leq |\mathcal{B}_\kappa|$, we allow each distinct price be the matching quality of the set for exactly one task, while only the highest price can be the matching quality of the team for more than one tasks. Any set of κ items that do not appear in the bids, are assumed to have a zero matching quality." Note that the above corresponds to an NOMTMT instance where the number of available agents are less than the number of required agents by all the tasks at hand, as described in [65]. The NOMTMT allocation problem has a feasible solution if and only if BCAWDP has a solution, which is shown to be $\mathcal{NP}$-hard. □

Typically, the winner determination problem for combinatorial auctions can be cast and solved as a linear program.

Table 7 General guideline on SUS score interpretation

SUS Score	Grade	Adjective Rating
> 80.3	A	Excellent
$68 - 80.3$	B	Good
68	C	Okay
$51 - 68$	D	Poor
< 51	F	Awful

Fig. 14 The CUQ questionnaire

	Strongly Disagree 1	Disagree 2	Neutral 3	Agree 4	Strongly Agree 5
The chatbot's personality was realistic and engaging	○	○	○	○	○
The chatbot seemed too robotic	○	○	○	○	○
The chatbot was welcoming during initial setup	○	○	○	○	○
The chatbot seemed very unfriendly	○	○	○	○	○
The chatbot explained its scope and purpose well	○	○	○	○	○
The chatbot gave no indication as to its purpose	○	○	○	○	○
The chatbot was easy to navigate	○	○	○	○	○
It would be easy to get confused when using the chatbot	○	○	○	○	○
The chatbot understood me well	○	○	○	○	○
The chatbot failed to recognise a lot of my inputs	○	○	○	○	○
Chatbot responses were useful, appropriate and informative	○	○	○	○	○
Chatbot responses were irrelevant	○	○	○	○	○
The chatbot coped well with any errors or mistakes	○	○	○	○	○
The chatbot seemed unable to handle any errors	○	○	○	○	○
The chatbot was very easy to use	○	○	○	○	○
The chatbot was very complex	○	○	○	○	○

Appendix C. Frontier of problem instance configurations that CPLEX can solve

The required time is average over five problem instances per combination ⟨number of tasks, number of agents, team size⟩ (Table 8).

Table 8 Frontier of problem instance configurations (number of tasks, number of agents, team size) along with the number of decision variables and required solving time

Number of Tasks	Number of Agents	Team Size	Number of Decision Variables	Required Time(sec)	Standard Deviation(sec)
10	100	2	49500	376.3446	17.45
20	100	2	99000	836.6631	17.29
30	100	2	148500	1153.8280	28.03
40	100	2	198000	1958.0188	74.93
50	100	2	247500	2305.5095	90.82
10	70	3	547400	4588.9023	74.45
20	60	3	684400	5505.4640	94.48
30	50	3	588000	5107.8094	181.27
40	40	3	395200	4045.5223	73.42
50	40	3	494000	4188.8293	179.38
10	30	4	274050	2386.0373	280.69
20	30	4	548100	5377.4400	408.94
30	20	4	145350	1328.7434	152.06
40	20	4	193800	1776.9613	193.64
50	20	4	242250	2228.5940	20.16
10	20	5	155040	1539.4960	243.25
20	20	5	310080	2906.0331	746.35
30	20	5	465120	6243.2947	61.33
40	20	5	620160	8438.8540	230.19
50	20	5	775200	8468.0339	130.67

References

1. Commission, E. A strategy for smart, sustainable and inclusive growth; Technical Report; European Commission, January. https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:52010DC2020&from=EN
2. Tino, C.; Grion, V. School-work alternation: Teachers' voices about the development and the assessment process of soft skills, In Trends in vocational education and training research. Proceedings of the European Conference on Educational Research (ECER), Vocational Education and Training Network (VETNET), 2018; pp 339–347
3. Wilde DJ (2009) Teamology: The Construction and Organization of Effective Teams. Springer-Verlag, London
4. Wilde, D. Post-Jungian Personality Theory for Individuals and Teams; SYDROSE LP, 2013
5. Andrejczuk, E.; Bistaffa, F.; Blum, C.; Rodríguez-Aguilar, J.A.; Sierra, C. Synergistic team composition: A computational approach to foster diversity in teams, Knowledge-Based Systems **2019**, 182 (104799)
6. Layton, R.A.; Loughry, M.L.; Ohland, M.W.; Ricco, G.D. Design and Validation of a Web-Based System for Assigning Members to Teams Using Instructor-Specified Criteria., Advances in Engineering Education **2010**, 2 (1), n1
7. Alberola JM, Del Val E, Sanchez-Anguix V, Palomares A, Teruel MD (2016) An artificial intelligence tool for heterogeneous team formation in the classroom. Knowledge-Based Systems 101:1–14
8. Purdue Univerity. CATME Smarter Teamwork. https://www.catme.org
9. IIIA-CSIC. Build teams with a diversity of genres, personalities and intelligences for more efficient collaboration. https://eduteams.iiia.csic.es/
10. Andrejczuk E, Rodríguez-Aguilar JA, Sierra C, Roig C, Parejo-Romero Y (2018) Don't leave anyone behind: Achieving team performance through diversity. In. IEEE Frontiers in Education Conference (FIE). IEEE 2018:1–9
11. Gibert A, Tozer WC, Westoby M (2017) Teamwork, Soft Skills, and Research Training. Trends in Ecology & Evolution 32(2):81–84
12. Cordis EU research results. A European AI On Demand Platform and Ecosystem (AI4EU). Grant agreement ID: 825619 https://cordis.europa.eu/project/id/825619
13. ESCO. European Skills, Competences, qualifications and Occupations, https://ec.europa.eu/esco/portal, 2010
14. IBM. IBM Ilog CPLEX Optimization Studio 12.10, https://www.ibm.com/us-en/marketplace/ibm-ilog-cplex
15. Cooper, A. The inmates are running the asylum. In Software-Ergonomie'99; Springer, 1999; pp 17–17
16. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching Word Vectors with Subword Information, arXiv preprint arXiv:1607.04606 **2016**
17. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space, arXiv preprint arXiv:1301.3781 **2013**
18. FBK. Competence and Skill Extraction component. Github repository https://github.com/dhfbk/document-classification
19. SAP. SAP Conversational AI | Low Code Chatbot Building Platform. https://cai.tools.sap
20. Wolf MJ, Miller K, Grodzinsky FS (2017) Why We Should Have Seen That Coming: Comments on Microsoft's Tay Experiment, and Wider Implications. SIGCAS Comput. Soc. 47(3):54–64
21. Capezzuto, L.; Tarapore, D.; Ramchurn, S.D. Anytime and Efficient Coalition Formation with Spatial and Temporal Constraints, 2020
22. Ponda, S.S.; Johnson, L.B.; Geramifard, A.; How, J.P. Cooperative Mission Planning for Multi-UAV Teams, Valavanis, K.P., Vacht-

sevanos, G.J., Eds.; Springer Netherlands: Dordrecht, 2015; pp 1447–1490
23. Anagnostopoulos, A.; Becchetti, L.; Castillo, C.; Gionis, A.; Leonardi, S. Power in Unity: Forming Teams in Large-Scale Community Systems, In , 01, 2010; pp 599–608
24. Lappas, T.; Liu, K.; Terzi, E. Finding a Team of Experts in Social Networks, In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA; KDD '09; Association for Computing Machinery, 2009; p 467-476
25. Anagnostopoulos, A.; Becchetti, L.; Castillo, C.; Gionis, A.; Leonardi, S. Online Team Formation in Social Networks, In Proceedings of the 21st International Conference on World Wide Web, New York, NY, USA; WWW '12; Association for Computing Machinery, 2012; p 839-848
26. Crawford, C.; Rahaman, Z.; Sen, S. Evaluating the Efficiency of Robust Team Formation Algorithms, In Autonomous Agents and Multiagent Systems, Cham; Osman, N., Sierra, C., Eds.; Springer International Publishing, 2016; pp 14–29
27. Andrejczuk, E. Artificial intelligence methods to support people management in organisations. Ph.D. Thesis, Artificial Intelligence Research Institute, CSIC, May, 2018
28. Andrejczuk E, Berger R, Rodríguez-Aguilar JA, Sierra C, Marín-Puchades V (2018) The composition and formation of effective teams: computer science meets organizational psychology. Knowledge Eng. Review 33:e17
29. Bachrach, Y.; Meir, R.; Jung, K.; Kohli, P. Coalitional Structure Generation in Skill Games, In , 01, 2010; Vol. 2
30. Czatnecki, E.; Dutta, A. Hedonic Coalition Formation for Task Allocation with Heterogeneous Robots, In 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC), 2019; pp 1024–1029
31. Präntare F, Heintz F (2020) An anytime algorithm for optimal simultaneous coalition structure generation and assignment. Autonomous Agents and Multi-Agent Systems 34(1):29
32. Chalkiadakis G, Boutilier C (2012) Sequentially optimal repeated coalition formation under uncertainty. Autonomous Agents and Multi-Agent Systems 24(3):441–484
33. O*NET. National Center for O*NET Development, www.onetonline.org/, 1990
34. SFIA. Skills Framework for the Information Age, https://ec.europa.eu/esco/portal, 2020
35. ISFOL. Professioni, Occupazione, Fabbisogni, https://fabbisogni.isfol.it, 2017
36. Li Y, Bandar ZA, Mclean D (2003) An approach for measuring semantic similarity between words using multiple information sources. IEEE Transactions on Knowledge and Data Engineering 15(4):871–882
37. Osman, N.; Sierra, C.; Mcneill, F.; Pane, J.; Debenham, J. Trust and Matching Algorithms for Selecting Suitable Agents, ACM Trans. Intell. Syst. Technol. **2014**, 5 (1), 16:1–16:39
38. Cohen SG, Bailey DE (1997) What makes teams work: Group effectiveness research from the shop floor to the executive suite. Journal of management 23(3):239–290
39. Kurtan, C.; Yolum, P.; Dastani, M. An Ideal Team Is More than a Team of Ideal Agents, In ECAI, 2020
40. Sandholm, T.; Suri, S.; Gilpin, A.; Levine, D. Winner Determination in Combinatorial Auction Generalizations, In Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: Part 1, New York, NY, USA; AAMAS '02; Association for Computing Machinery, 2002; p 69-76
41. GUROBI. Gurobi Optimizer 8.0, https://www.gurobi.com/
42. GLPK. GLPK: GNU Linear Programming Kit), https://www.gnu.org/software/glpk/
43. Gamrath, G.; Anderson, D.; Bestuzheva, K.; Chen, W.K.; Eifler, L.; Gasse, M.; Gemander, P.; Gleixner, A.; Gottwald, L.; Halbig, K.; et al. The SCIP Optimization Suite 7.0; Technical Report; Optimization Online, March, 2020
44. Morrison RW, De Jong KA (2002) Measurement of Population Diversity. In: Collet P, Fonlupt C, Hao JK, Lutton E, Schoenauer M (eds) Artificial Evolution, Berlin, Heidelberg. Springer, Berlin Heidelberg, pp 31–41
45. Palmero Aprosio, A.; Moretti, G. Tint 2.0: an All-inclusive Suite for NLP in Italian **2018**, 10, 12
46. Joulin, A.; Grave, E.; Bojanowski, P.; Mikolov, T. Bag of Tricks for Efficient Text Classification, arXiv preprint arXiv:1607.01759 **2016**
47. Jones KS (1972) A statistical interpretation of term specificity and its application in retrieval. Journal of Documentation 28:11–21
48. Zhou T, Wang Y, Zheng X (2020) Chinese text classification method using FastText and term frequency-inverse document frequency optimization 1693:012121
49. Conitzer, V.; Sandholm, T. Complexity of Manipulating Elections with Few Candidates, In Eighteenth National Conference on Artificial Intelligence, USA; American Association for Artificial Intelligence, 2002; p 314-319
50. Faliszewski, P.; Hemaspaandra, E.; Hemaspaandra, L.A.; Rothe, J. Llull and Copeland Voting Broadly Resist Bribery and Control, In ; AAAI'07; AAAI Press, 2007; p 724-730
51. Council of European Union. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation), 2016. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32016R0679&qid=1639589889181
52. Proposal for a Regulation laying down harmonised rules on artificial intelligence (Artificial Intelligence Act), May, 2021. https://digital-strategy.ec.europa.eu/en/library/proposal-regulation-laying-down-harmonised-rules-artificial-intelligence-artificial-intelligence
53. "High-Level Expert Group on Artificial Intelligence". Assessment List for Trustworthy Artificial Intelligence, July, 2020. https://digital-strategy.ec.europa.eu/en/library/assessment-list-trustworthy-artificial-intelligence-altai-self-assessment
54. Dignum, V.; Nieves, J.C.; Theodorou, A.; Tubella, A.A. An abbreviated assessment list to support the Responsible Development and Use of AI, April, 2021
55. Zillner, S. Innovation in Times of Big Data and AI: Introducing the Data-Driven Innovation (DDI) Framework, Curry, E., Metzger, A., Zillner, S., Pazzaglia, J.C., García Robles, A., Eds.; Springer International Publishing: Cham, 2021; pp 289–310
56. Crowd4SDG. Crowdsourcing for Sustainable Development Goals: a Data Repository and a Toolbox for Citizen-Science Communities. A Horizon 2020 Research and Innovation Action, https://crowd4sdg.eu/. Last Accessed: 2023-05-01
57. YOMA. YOMA: an online marketplace for youth which provides opportunities to develop skills, engage with a community and access employment., https://www.yoma.africa/. Last Accessed: 2023-05-01
58. Wilde, D.J.; et al. Teamology: the construction and organization of effective teams; Springer, 2009; Vol. 3
59. Moreno JL, Jennings HH (1938) Statistics of Social Configurations. Sociometry 1(3/4):342–374
60. Ballesteros-Perez P, González-Cruz MC, Fernández-Diego M (2012) Human resource allocation management in multiple projects using sociometric techniques. International Journal of Project Management 30(8):901–913
61. Fiol CM (1994) Consensus, diversity, and learning in organizations. Organization science 5(3):403–420
62. Miller T (2019) Explanation in artificial intelligence: Insights from the social sciences. Artificial Intelligence 267:1–38

63. Holmes, S.; Moorhead, A.; Bond, R.; Zheng, H.; Coates, V.; McTear, M. Usability testing of a healthcare chatbot: Can we use conventional methods to assess conversational user interfaces?, In Proceedings of the 31st European Conference on Cognitive Ergonomics, 2019; pp 207–214
64. Brooke, J. Sus: a "quick and dirty'usability, Usability evaluation in industry **1996**, 189
65. Georgara, A.; Rodríguez-Aguilar, J.A.; Sierra, C. Towards a Competence-Based Approach to Allocate Teams to Tasks, In Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems, Richland, SC; AAMAS '21; International Foundation for Autonomous Agents and Multiagent Systems, 2021; p 1504–1506