U.P.B. Sci. Bull., Series C, Vol. 86, Iss. 1, 2024 ISSN 2286-3540

TOWARDS A RECOMMENDATION SYSTEM FOR AN EDUCATIONAL PROFILE IN SYSTEMS ENGINEERING

Veronica OPRANESCU[1], Anca Daniela IONITA[2]

Students need to personalize their educational profile and usually find disparate information from various sources. Our aim is to integrate smart campus concepts and provide a recommendation system for bachelor students, using cloud services. The resulted assistant uses a chatbot to provide decision support and recommend the most appropriate specialization according to the students' profile, as well as a long-term educational, including ideas for master's programmes to continue their educational journey. The training was done by collecting data from real users and also by synthetic data generation.

Keywords: engineering education, chatbot, recommendation system

1. Introduction

A smart campus includes a large variety of technologies to improve students' experience and give them access to a new variety of services, like freshmen orientation [1] and automation of administrative processes [2]. The increasingly prevalent emergence of recommendation systems in the context of smart campuses has also been identified, highlighting the undeniable benefits they bring to all stakeholders involved [3]. Our research is focused on a recommendation system to guide students throughout the process of choosing an educational profile, starting from the third year of their bachelor's degree. It assists in the selection of a suitable specialization for the fourth year and continues until the master's degree stage. The application is designed for a specific group of users, i.e., students in their final years of study in our faculty, specializing in Systems Engineering. It aims to help them make informed decisions regarding their specialization within their current field of study and guide them towards a recommended master's programme. Section 2 includes the description and the comparison of existing platforms that provide support for the development of a recommendation system. Section 3 describes the steps executed in order to develop a recommendation system for our scope and Section 4 presents the performance obtained.

2. Background

Recommendation systems have applications in various domains, with commonly acknowledged instances including product recommenders for online

[1] PhD student Eng., Automation and Industrial Informatics Dept., National University of Science and Technology POLITEHNICA Bucharest, Romania, e-mail: veronica.opranescu@yahoo.com
[2] Prof., Automation and Industrial Informatics Dept., National University of Science and Technology POLITEHNICA Bucharest, Romania, e-mail: anca.ionita@upb.ro

stores, content suggestions for all types of social media platforms and similar collections, consisting of playlists for video and music streaming services [4]. In terms of approaches, most recommendation systems use both collaborative filtering and content-based filtering, referred to in the specialized literature as personality-based approach and knowledge-based systems.

Collaborative filtering methods construct models based on user's historical interactions (such as items previously viewed or selected, and numerical ratings assigned to those items), as well as analogous choices made by other users [5]. This type of method relies on the assumption that individuals who have agreed on a certain process/item/subject in the past, will continue to agree in the future and will prefer similar types as those they favored in the past. Using this kind of approach, the system generates recommendations based solely on the evaluation profiles of various users, or process/item/subject. Through the identification of elements sharing similar existing records with the present user or item, these systems derive recommendations by exploiting this observed pattern. Collaborative filtering methodologies are commonly divided into two main categories: memory-based and model-based approaches [6].

On the other hand, if we analyze the approaches focused on content filtering, these are based on gathering information about certain users, in terms of preferred elements and creating their relevant profiles, having limited information about the students [7]. The main difference is the method that content-based recommendation systems use to create a collection of suggestions. To provide a better understanding of this process, the recommendation system plays the role of a classification task, designed especially for the user. One important step is the user’s profile generation process, that combines his/her preferred elements in terms of content. It is crucial to distinguish the likes/dislikes for a specific user in order to create a relevant and accurate profile.

Our research aimed to use a chatbot for transmitting recommendations to the users. During the selection process for the most suitable hosting service for a chatbot application, careful consideration was given to, and thorough analysis was conducted on five existing platforms: Dialogflow, Amazon Lex, IBM Watson, Wit.ai and Azure Bot Service. To ensure an appropriate selection process, a set of defining criteria was taken into account for the implementation of the application and analyzed in Table 1.

There are two main categories of chatbot solutions: (i) the classic one, as known as rule-based, which is more limited but cost-effective, can be developed rapidly and ensure predictable levels of efficiency and customer satisfaction, and (ii) the AI-powered solutions, which are capable of handling complex requests, but are more expensive and require higher learning effort [8].

Table 1

Recommendation platforms' assessment

	Dialogflow	**Amazon Lex**	**IBM Watson**	**Azure Bot Service**
Channels	Voice, Text	Voice, Text	Voice, Text	Voice, Text
Ease of use	It offers a web-based interface for bots creation, simplifying the process for developers of all skill levels.	Furnishes a web-based platform for bot creation and deployment, using the same machine learning engine as Alexa.	Delivers a user-friendly interface that's intuitive and comes equipped with video tutorials and readily available sample materials.	Frequently used for crafting intelligent bots, it boasts an integrated web interface designed to facilitate the publishing process.
Integrations	-Google Assistant - Slack - Facebook - Twitter	- Slack - SMS - Facebook - Twillo	- Voice Agent - Slack - Facebook - Wordpress	- Facebook - Skype - Slack - Telegram
Web and Mobile Integrations	Provides basic in-built web integra-tion and codeless integration with Kommunicate.	Provides basic chat UI for website testing.	Provides basic UI for websites.	Open source web chat widget available in Github
Languages	20+ languages, including English, French, Spanish.	Only English	10+ languages, including English, Chinese, Japanese.	Supports English, Spanish, German, French
Cost	Free standard plan or Enterprise version ($0.002/request).	Free: 10k text +5k speech req, then $0.004 voice or $0.00075 text req.	Free: 10k messages/ month + restrictions Paid plan: $0.0025/ message.	Free: 10k requests/ month Paid plan: $0.5/ 1000 messages.

All platforms mentioned and analyzed before (Dialogflow, Amazon Lex, IBM Watson, and Azure Bot Service), support both types of chatbots:

(i) Rule-based chatbots – the simplest form of intelligent assistants that can be developed. As the name suggests, they rely on a set of predefined conditions and rules to function. In terms of language, the rules can range from simple to highly complex, covering various topics from multiple domains.

(ii) AI-powered chatbots – the enhanced and advanced version of virtual assistants, leveraging a wide range of modern and untapped technologies, such as Natural Language Processing (NLP) and Machine Learning (ML). One noticeable difference from rule-based chatbots is the category to which they belong. AI-based chatbots have the capability to continuously train and improve throughout their lifecycle and interaction with users. Through the training process, the performance of AI-based chatbots constantly enhances.

3. Educational Profile Recommendation System

3.1. Select the recommendation platform

We selected the IBM Cloud development environment, Watson Assistant, to check certain usage requirements. One of the most significant requirement is the conversational use of the English language within chatbot conversations. While initially this condition may seem like a disadvantage for a faculty whose courses are conducted in Romanian, a closer analysis reveals that due to the study domains within our faculty (Systems Engineering, Automation and Applied Informatics), the use of the English language is not just a recommendation, but a professional necessity. Furthermore, some undergraduate students apply to master's or doctoral programmes at universities in other countries, which requires knowledge of the field of study in an international language, usually English. Additionally, for future employment in this field, students will be part of international corporations where specialized vocabulary in English is essential. Developing a chatbot in an internationally used language represents a step towards globalization and the integration of international students [9].

To develop a recommendation application based on intelligent assistants, IBM Watson Assistant service was used, within which the programmer can define the dialogue flow, entities, and intents. This plan is designed for educational purposes, allowing the creation, testing, and publishing of the assistant. The training process of the chatbot depends on the dataset available and the complexity of the conversation flow defined.

3.2. Method for defining the student educational profile

Considering the relevant information regarding the role of models in software project development, the first step approached is the definition of the student educational profile. The aim is to abstract the studied object in order to use this concept across different interfaces and stages of the project. Thus, the student object is analyzed based on the characteristics and relationships of interest for the virtual assistant.

By utilizing a Google Forms quiz questionnaire, data was collected from real users (students of our faculty) using a Purposive Sampling method, more specifically, Homogeneous Sampling method, where individuals (65 participants) were deliberately chosen from a larger group [10] (the students of our faculty, who were in the position of choosing a specialization before the final year or even a master's programme) to obtain targeted and insightful information. Unlike Convenience Sampling method, it relies on the researcher's judgment and predefined criteria, making it ideal for in-depth and specialized research inquiries, as this one. The questions addressed by the students varied, with diverse formulations, being asked to address realistic inquiries, specifically related to the scope of this research.

Among these, the following have been identified:

- The field and direction they are currently studying
- Subjects the student wishes to study the following year
- Topics/subjects of interest for future studies (master's programme)
- The characteristic subjects for each specialization/branch, within each direction (For Systems Engineering [Ingineria Sistemelor]/IS, there are: Direction A – specializations: A1, A2, A3, Direction B – specializations: B1, B2, B3 and for Computers and Information Technology [Calculatoare si Tehnologia Informatiei]/CTI with specializations: C1, C2,C3,C4,C5)
- Keywords that indicate specific topics/subjects of interest, both for the next year and for the master's programme.

In order to retrieve accurate data and correct names, the National Register of Qualifications in Higher Education (RNCIS) was consulted, choosing Faculty of Automation and Computer Science, for the bachelor's degree study cycle, in the field of Systems Engineering, and the program of study being Automation and Applied Informatics. All subjects are translated into English, making them suitable for chatbot usage. The information provided by students regarding the topics of interest in the last three years of study include concepts and specific programs related to each subject, enabling the chatbot to easily identify the described domains. Furthermore, data regarding the final year of study for the bachelor's degree were introduced, as the student will need to choose a specialization based on the direction he/she follows in the third year [11][12].

3.3. Dynamic allocation of questions and answers in the chatbot

Dynamic allocation of answers is performed through a scheduled programmatic call using a webhook, which sends a POST request to an external application that executes the programmed function. The webhook receives a set of pre-defined instructions within a JSON file. The webhook returns the payload of the JSON object, which is then used by Watson Assistant.

The webhook was used as a mechanism that allows calling an external program based on events identified in the chatbot's workflow. Within a dialog skill, the webhook was triggered when the assistant processed a node where the webhook is activated. The webhook collects the data specified in the JSON file, or provided by the user, and saves them into context variables. It then sends the data as part of an HTTP POST request to the URL specified in the webhook definition.

Entities have been defined for the chatbot, in order to identify certain components in the input received from the student, which will help build the educational profile. Among these entities are the types of subjects specific to each specialization and each master's programme. The next step was to define each subject and a group of synonyms for it, in case the user does not enter the exact term. The questions that the chatbot will ask are aimed at shaping the student's profile. A series of guiding questions have been defined using a general model, such as "What are your favorite subjects among those you have studied so far?", "What topics would you be interested in studying further?" etc.

3.4. Recommendation system functionalities

In order to provide an educational path recommendation based on the specialization chosen in the final year of the undergraduate programme, the intelligent assistant needs a collection of data to build the student's profile, subsequently serving as the basis for the received suggestion. The student needs to provide as input the subjects/topics of interest from the fourth year of the undergraduate cycle. Each specialization belonging to both directions (A and B) has been defined in the Watson Assistant chatbot dictionary, along with a series of words/terms that have a similar meaning to that subject or are related to its lexical field. Each treated specialization has been treated as an entity, and each specific subject as a possible value of that entity, along with the corresponding collection of synonyms.

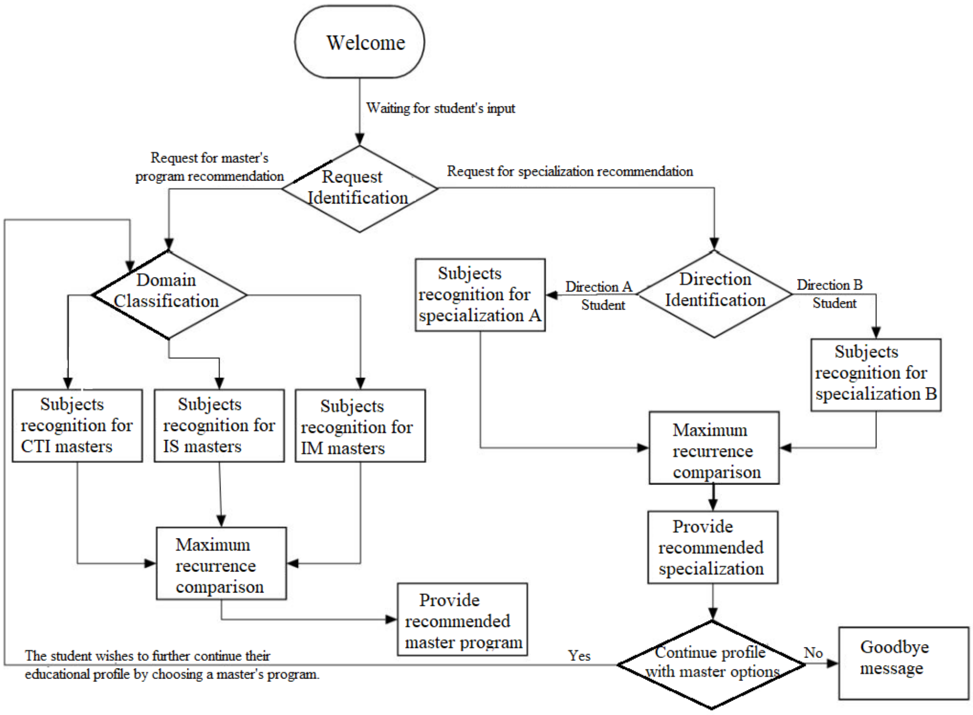

Fig.1. Process flow for chatbot

This information has been extracted into a .csv document and loaded into the designated space for the dictionary of each entity.

In case of choosing subjects dedicated to a specialization belonging to Direction A, the student's input will be analyzed, and the existing references will be checked and stored in context variables. Based on the recurrence of specifying subjects characteristic to a specialization, the recommendation will be selected by choosing the package with the highest number of preferences. At this moment, the recommendation system is capable of defining the student's preference profile and, based on that, of executing the next steps in its logical flow (see Fig. 1).

The student will be later asked if he/she is interested in pursuing further studies in a master's programme, and in case of a positive response, additional input will be requested regarding their areas of interest. Using NLP, key words will be identified for each concept, and the options will be recognized as the student's preferences. One assumed that the user would enter any number of preferred subjects, so the assistant would be able to provide a specialized response based on the frequency of characteristic subjects for each specialization. For example, if the student enters multiple subjects specific to A1 specialization, the chatbot will display the following recommendation message: "Considering the subjects you are interested in, I recommend you choose the A1 specialization." Other cases were also handled, where the student inputs lead the chatbot to recommend two or three specializations ("It seems you want to study subjects that are specific to both A1 and A2 specializations. You can restart this recommendation process and insert other subjects for a more precise option.").

The Fuzzy Match feature provided by Watson Assistant has been activated to enhance the chatbot's ability to recognize entities even when they are misspelled or incorrectly written by students. The final recommendation reached by the assistant is transmitted to the student, offering them the possibility to continue their educational profile, by choosing a suitable master's programme, according to the preferences. The chatbot will analyze the student's request, and if the response is positive, it will proceed with the scenario of recommending a master's programme. The first step in this process is to identify the field in which the student wants to pursue their studies (CTI, IS, or IM/ Engineering and Management [Inginerie și Management]), and they will be asked to provide topics of interest that they wish to study in the future as input. Based on the entities recognized by the NLP algorithm, the recommended option for continuing their studies at the master's level is proposed.

3.5. Training process

The dataset was composed using the two types of methods, as no existing dataset could be found to map the requirements and needs identified by the chatbot;

the analyzed subjects/topics are specific to the Faculty of Automatic Control and Computers.

Collecting data from real users. Results obtained from distributing the questionnaire among our faculty students were used, within which 65 persons, that share their position as a undergraduate in final year for the bachelor degree interested in choosing a specialization and for the future (or graduated students), a master's programme, completed the quiz. These results were saved in a CSV file and later utilized in the training process. In this stage of the implementation, the dataset is structured in a single column, containing all the inputs provided by the students in the form of English sentences. In the next phases, this dataset is divided into 80% for training phase and 20% for testing phase.

Synthetic data generation. This means creating datasets using a computer, to increase the size of the initial dataset or introduce specific information that the model should process in the future. The dataset obtained using the previous method has a relatively small scale (65 students participated within quiz' completion process, within which a total number of 12 questions were addressed), and it is necessary to increase the number of records in order to train the machine learning model effectively. Synthetic data generation provides a flexible and cost-effective alternative in terms of effort and time, compared to collecting a large amount of real data. Generative Adversarial Networks (GANs) is an advanced technique used in the Python notebook, which generates synthetic data that resulted in tripled number of dataset' elements (195 synthetic entries).

To train the virtual assistant developed using IBM Cloud Watson Assistant, several aspects were taken into consideration, among which the training process dedicated to NLP elements. For a recommendation system such as the one developed, in order to train NLP part, it was mandatory to implement a supervised machine learning technique. It was identified the necessity to map the input received from user to existing and documented entities, such as subjects, specializations, and master's programmes. The initial dataset provided training data, which was tagged and categorized in the next phase, thereby creating a mapping system for the output [13].

The machine learning components integrated with the chatbot include methods for data prediction and classification. SVM (support vector machine) method stands out, which was enabled upon switching on the "Plus trial" plan, in order to improve solution generation in case the student requires additional suggestions regarding a specialization or a master's programme. Of course, there are several variants derived from deep learning [14], such as neural networks, which could have aided in the training process, but the SVM method remains one of the most robust solutions for classifying statement correspondences with defined intents. During the training process, the SVM method refined the suggestions received from students who had access to the distributed questionnaire, executing additional steps, like filtering out the entries that represented potential weak points for the features available within NLP usage.

4. Performance analysis and discussion

The main purpose of this stage is to understand the shortcomings of the chatbot and what has worked according to the initial plan. To assess the performance of the recommendation system, measurements were used for the two relevant features: efficiency and coverage, and the combination of them provides a powerful tool for diagnosing the developed system [15]. If both the efficiency and coverage metrics have higher values, it can be concluded that the analyzed chatbot responds positively to all requests. Conversely, if the values of both features are low, the metrics provide helpful information to improve the performance of the chatbot.

The coverage is expressed as a percentage and represents the extent to which the chatbot is able to provide valid responses to user messages. It can be measured by considering the total number of messages for which the chatbot returns a valid solution. The coverage value can be measured using the boundaries set by the defined intents and queried in the dialog nodes.

If a received message falls below the threshold set by the predefined intent (0.20) and is marked as irrelevant, it will not be processed. However, if during the performance analysis one notices that users consistently ask questions that the chatbot has not been trained to answer, a series of conversations covering those types of questions can be included in the training dataset.

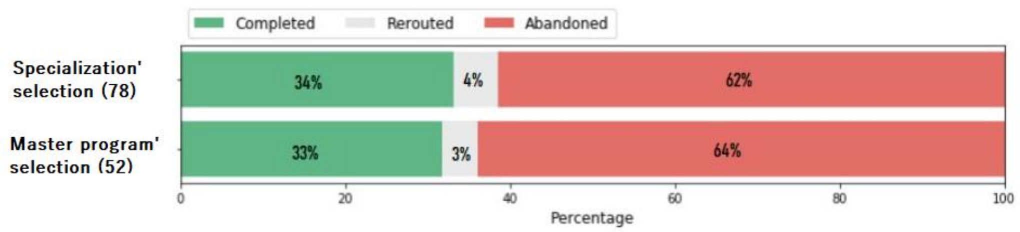

Fig.2. Efficiency metrics for task completion

The results for efficiency metrics regarding task completion are presented in Fig. 2, based on a total number of 130 interactions with real users (targeted group of students), among which 78 conversations were tagged as Specialization' selection and the remaining 52 conversations were classified as Master program' selection. The "Abandoned" state refers to a conversation that terminated in the middle of the flow. "Rerouted" refers to a conversation that left the scope of the flow and did not return. "Completed" refers to conversations that successfully reached the completion point.

High abandonment values occur due to complex or unclear responses (if the chatbot provides this type of responses, users may become frustrated and abandon the conversation), lack of personalization, limited knowledge (if the chatbot lacks sufficient knowledge or frequently responds with "I am not trained for this subject" to user queries), ineffective Natural Language Understanding (the chatbot misinterprets user queries or fails to understand context), long response delays, absence of human handoff and insufficient user guidance (users might not know how to use the chatbot effectively or what types of queries it can handle). To improve these values in future versions, it is crucial to:

- improve user context tracking and personalization so that the chatbot can refer to past interactions, user preferences;
- continuously update and expand the chatbot's knowledge base by adding more relevant content and training it with new data (integrating external sources of information is also an option).

These strategies collectively help create a more engaging and user-friendly chatbot experience, reducing abandonment and increasing user satisfaction.

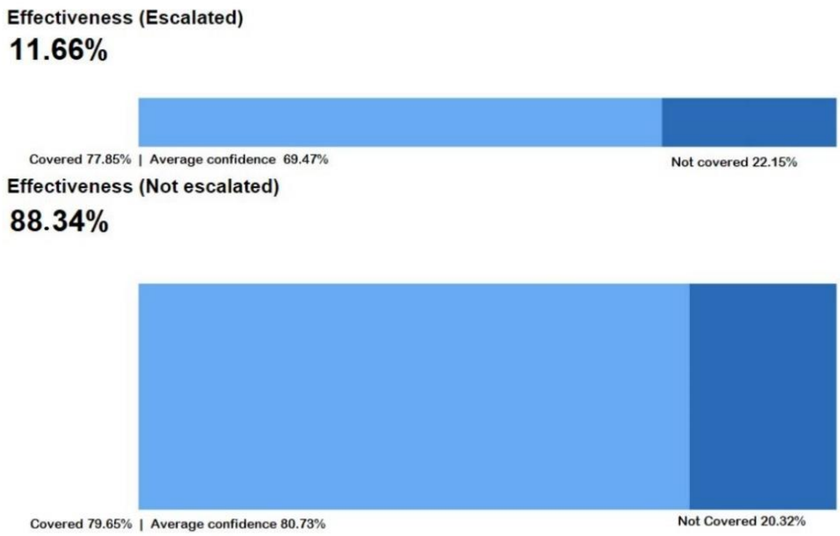

Fig.3.Metrics for quality (efficiency) and coverage after the training process

Quantifying the effectiveness (quality) and coverage of a chatbot are complementary activities that help to gain a better overall understanding of the entire system [16]. If the virtual assistant provides incorrect answers to student questions, improving effectiveness (quality) can address this disadvantage. Using the Python script developed in the Watson Assistant Measure Notebook, a relevant analysis of effectiveness and coverage values was performed.

Before the training process, 32.42% of conversations were escalated to a simulated agent, compared to 11.66% obtained after the training (see Fig. 3). Furthermore, within these escalated conversations, the assistant has responded to a significant number of questions (56.11%), which suggests a current improvement needed in regard with the effectiveness of the responses; there is not a lack of defined responses (i.e., coverage), so after the training, the results improved significantly, to 77.85%.

There have been several limitations that influenced the performance of the current version of recommendation system like: small-sized training and test datasets, limited interaction time with real users. The initially acquired plan was the Lite plan (completely free), which had several restrictions, including limitations on training methods. It was therefore necessary to apply for a “Plus trial” plan, with a shorter duration of use (for this application, it was used for 30 days). The “Plus trial” plan granted access to advanced features such as expanded conversation limits, machine learning tools, and enhanced scalability, making it suitable for improving the user experience. It provided valuable insights through comprehensive analytics and improved response times. In terms of system performance, the use of a “Plus trial” plan has led to **better accuracy** (by advanced machine learning tools and natural language understanding features) generating an increase in addressing user queries processes and **enhanced customization** (greater flexibility and customization options) enabling the developer to tailor the chatbot's behavior and responses.

5. Conclusions

The implemented recommendation system falls within the scope of the smart campus context, this recommendation system aiming to guide students throughout the process of personalizing an educational profile. The application is intended for the final year students of the our faculty in the field of Systems Engineering, who are about to make their specialization choice based on the educational direction they are already studying in, as well as guiding them towards a recommended master’s programme.

To improve the performance of the system, the training process was employed, implemented in two distinct stages. The first stage involved collecting the necessary training dataset (corpus) from real users (65 participants) using the distributed questionnaire and training the chatbot with these queries. Measurements of the recommendation performances that the assistant was trained on were performed (based on a total number of 130 interactions with real users), and the obtained graphs were retained, in order to compare and visualize their improvement after the next training stage. It was observed that the size of resulted dataset was insufficient, so its scale extended 3 times using Synthetic Data Generation and due to Plan trial selection as a training option, better accuracy and enhanced customization were achieved, thereby exceeding the challenges given by the reduced dataset and limited interactions with real users, decreasing the escalation percentage with 20%.

In terms of future work, several additions are targeted to enhance the performance of the recommendation system, such as: improving the algorithm for generating synthetic data, utilizing a crowdsourcing approach to collect data from a significantly larger number of real users, and adopting a web scraping technology to develop subject-specific dictionaries for each specialization/ master's programme.

R E F E R E N C E S

[1]. *I. Damian, A.D. Ionita, S.O. Anton*, "Community- and Data-Driven Services for Multi-Policy Pedestrian Routing, Sensors", **vol 22**, issue 12, art. no. 4515.

[2]. *V. Opranescu, I. Nedelcu, A.D. Ionita*, "Automating Students' Decision Processes in a Smart Campus", 13th International Symposium on Advanced Topics in Electrical Engineering (ATEE), Bucharest, Romania, 2023, IEEE, pp. 1-6

[3]. *V.Ahmed, K.A.Alnaaj, S.Saboor*, "An Investigation into Stakeholders' Perception of Smart Campus Criteria: The American University of Sharjah as a Case Study", Department of Industrial Engineering, American University of Sharjah, Sharjah, Sustainability, 2020.

[4]. *A.Dyck, A.Ganserand, H.Lichter*, "A Framework for Model Recommenders", SCITEPRESS (Science and Technology Publications), 2014.

[5]. *J.B.Schafer, D.Frankowski, J.Herlocker, S.Sen*, "Collaborative Filtering Recommender Systems", The Adaptive Web. Lecture Notes, Computer Science, **vol 4321**, 2007.

[6]. *A.Bucchiarone, J.Cabot, R.F.Paige*, "Grand challenges in model driven engineering: an analysis of the state of the research", in Springer, 2020.

[7]. *N. Karbhari, A. Deshmukh and V. D. Shinde*, "Recommendation system using content filtering: A case study for college campus placement", International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), 2017, pp. 963-965.

[8]. *J. Jia*, "CSIEC: A computer assisted English learning chatbot based on textual knowledge and reasoning", in Knowledge-Based Systems, no. 22, 2019, pp. 249–255.

[9]. *J.Feine, U.Gnewuch, S.Morana, A.Maedche*, "A taxonomy of social cues for conversational agents", in International Journal of Human-Computer Studies, no. 132, 2019, pp. 138–161.

[10]. *I.Etikan, S.A.Musa, R.S.Alkassim*, "Comparison of Convenience Sampling and Purposive Sampling", in American Journal of Theoretical and Applied Statistics, **vol. 5**, 2016, pp. 1-4.

[11]. "Educational plan for Computers and Information Technology Program." Acs.pub.ro. https://acs.pub.ro/public/Plan-de-invatamant-Calculatoare-Licenta-2020-2021.pdf/ (accessed May 25, 2023)

[12]. "Educational plan for Systems Engineering Program." Acs.pub.ro. https://acs.pub.ro/public/03_AC_L_2020-IS.pdf/ (accessed May 25, 2023)

[13]. *K.H.Tae, Y.Roh, Y.H.Oh, H.Kim, S.E.Whang*, "Data Cleaning for Accurate, Fair, and Robust Models: A Big Data – AI Integration Approach", DEEM '19, 2019

[14]. *S.V.Teja*, "Chatbot using deep learning", in Academic Leadership-Online Journal, no. 21, 2020, pp. 428–438.

[15]. *V.J.J. Flores, O.J.J. Flores*, "Performance Comparison of Natural Language Understanding Engines in the Educational Domain", IJACSA (International Journal of Advanced Computer Science and Applications), **vol. 11**, 2020, pp. 8.

[16]. *D.Braun, A.H.Mendez*, "Evaluating Natural Language Understanding Services for Conversational Question Answering Systems", SIGDIAL Conference, 2017, pp 174-185.