Design of an Educational Virtual Assistant Software

https://doi.org/10.3991/ijet.v16i11.19847

Dominik Halvoník (✉)
Department of Informatics Constantine the Philosopher University in Nitra,
Nitra, Slovakia
dhalvonik@ukf.sk

Peter Psenak
Comenius University, Bratislava, Slovakia

Abstract—Despite the great creators' efforts of e-learning educational materials, it is not possible to define the content of these materials specifically for all students. Based on this, it is necessary to provide in e-learning education the possibility of a more detailed interpretation of specific parts of the educational material that may be unclear to specific students. Based on this fact, we decided to take the first step in the form of software design, which will represent a virtual assistant in teaching computer science. The role of this assistant will be the ability to answer technical questions related to the presented curriculum. From an architectural point of view, it will be a set of micro-services, each of which will serve a specific task. The prerequisite is the use of decision trees to determine a specific micro-service, which will be implemented in the form of a neural network. The main aim of this paper is to provide a detail description of global software architecture for such a virtual assistant.

Keywords—E-learning, virtual assistant, decision tree, neural network, software design

1 Introduction

The basic parameter of successful education is the understanding of the presented curriculum. In education through e-learning, it is relatively difficult to actively enter the educational process with the position of a teacher. This fact has been shown quite clearly by the current ongoing distance learning in several EU countries caused by the Covid-19 pandemic. There are several ways in which a teacher can positively influence a student even if the teacher is not actively present in the educational process. We are talking, for example, about the appropriate structuring of educational materials, whether in terms of content or form [1]. Nevertheless, it must be said that it is not uncommon for a student to simply "get stuck" on a certain part of the educational content and not be able to continue without assistance [2]. If he is not able to get an explanation of the subject himself, a situation arises where the student is unable to continue learning and must wait for the teacher to explain the problematic topic.

We assume that due to the time variability in e-learning education, a teacher or assistant cannot be available for every student. Therefore, eliminating this problem is, from our point of view, feasible by creating and integrating a virtual assistant who would be able to answer students' questions at a time when the teaching process is underway. At the same time, it must be said that the assistant should have the ability not only to replicate learned answers but also to create new ones based on interaction with students. We would perhaps best meet these assumptions when using neural networks. The problem, however, is that the neural network will most likely not be able to generate the correct answers if there is an overlap of topics [3]. An example of such a problem is an attempt to expand the scope of knowledge of said assistant from information about the subject of informatics to the subject of biology. In that case, if a student comes up with a question containing the word "Python", for example: “What is python?”, it is very likely that the neural network can provide an answer from both thematic units. This increases the risk of a bad answer. We would like to solve this fact by creating several specialized neural networks that will operate independently and will be divided based on the content in which they can assist. In this paper we present main findings in structured form:

- Global overview of virtual assistant architecture from software perspective
- Explanation of experimental results which leads to specific software component selection
- A detailed description of the logical concept of the presented virtual assistant based on preliminary research
- Pointing out the pitfalls of the presented concept and the solutions to these challenges

2 Literature Review

During the last decade, we witnessed the rise of chatbot usage in organizations everywhere in the world. They have improved the everyday life of people as well as the performances of the business sector. This was achieved by chatbots, and their ability to make more effective communication with customers worldwide [4]. We can chat with a computer to order a pizza or to buy coffee for a specific time. We can write to a “restaurant” to reserve a place at an exact time. Furthermore, Alexa, Siri, or other personal assistant technologies can be found nearly in every household. We are capable to manage lights at home from our mobile phones.

There are some differences between chatbots and virtual assistants. Chatbots cannot communicate only in a linear fashion, therefore they are incapable of accepting any user feedbacks [5]. This ability can be found in virtual assistants. They both utilize the same technologies and algorithms of Natural Language Processing, Machine Learning, and/or deep learning. The combination of these tools enables them to work with people, check their emotions, use search engines in cases when they are prompted to or form responses that are more user-orientated. We would like to use these specific abilities or functions in education, to enhance the value it.

The main tool for interactivity of the virtual assistants is NLP which is a specific field of science for designing methods and algorithms that work with unstructured data. Specifically, the input or the output of the different algorithms produces unstructured sequences of data. Because the human language is highly ambiguous and can be highly variable in different languages the algorithms must consider the specific differences and be ready to respond to them in the most normalized and standardized manner [6]. This, in our case comes with a problem in the Slovak language, which is not so thoroughly analyzed as English.

The human language is ever-changing and evolving through the times which must be considered in the models. In the case of people, the human language is greatly understandable and even the small nuances in the meaning of words can be easy to interpret. It is different in the case of computers. They are less sophisticated in realizing the underlying nuances but capable of formally understand the languages even better than humans. This problem of computers understanding spoken words is the base problem of any NLP project [6].

Computer-aided education gives as the opportunity to create software to automate the evaluation of students' tasks [7] or to create interactive materials for better student-teacher interaction. In the case of virtual assistants, the teacher can work with a multitude of new approaches to classes. Students who are considered to be digital natives have slowly changed their learning processes. They can get information quicker using mobile devices and this can be used for their learning process as well [20]. The technology can help further us in university education [19].

Chatbots are useful for multiple reasons. First, they can respond immediately and anytime which is useful in case of student interaction. If a student would like to get an answer to their question after school time, he can get it instantly from a chatbot. This can be a great asset for students because they can forget the question until they meet the teacher again. A student's question should be answered empathetically; this depends only on the programming of the chatbots. However, a good and friendly answer can motivate the students to ask more questions and further enrich their knowledge base [8]. Another helpful feature of virtual assistants is the customizable experience, which they are capable to create. They do it by adjusting teaching methods based upon the individual students. This may allow the students to focus more on special areas which they need to improve more. These virtual assistants can therefore set up their communication style specifically for each student [9]. Furthermore, chatbots and virtual assistants can have a positive impact on teachers' life as well. They may help in assignments, exams or only with answering to common questions which the teachers repeat all the time [10]. Therefore, these technologies can fill the role of teaching assistants and they can achieve it in a faster fashion with fewer mistakes.

Different Universities utilize chatbots and Virtual assistants for the last couple of years. There is a specific one at Staffordshire University called Beacon. It can check the students during tests, give them reminders, or recommend specific after school activities. It is capable to give them basic answers regarding the places around the campus or can be a bridge between school administrators and parents, making it easier to communicate with each other. This case study can be also applied on a much bigger scale [11] which provides us with proof that the solution applies to much larger and more computationally demanding tasks.

We have to consider the readiness of the teachers for interactive tools. It is still a high demanded research question of how to utilize and prepare the pedagogical personnel to use these specific technologies. Using interactive applications requires multiple changes in the teacher behavior in classes, their approaches to pedagogical work, preparation and realization of teaching the different subject material as well. The teacher's knowledge of information technologies and the skills in using computer-assisted techniques during the pedagogical processes are often unsatisfactory. We have to create a virtual assistant which is easy to use so even the pupils or students would be able to work with the assistant and the teachers even with a lower level of informatics knowledge would be able to utilize the technology and help achieve higher interactivity during classes.

Neural networks are able to provide the needed capabilities to resolve natural language-based problems. The most prominent component of neural networks for language is the inner use of embedding layers. With this, the network maps discrete symbols into continuous vectors in an admittedly low dimensional space. With this transformation in place, they transform into mathematical objects on which future calculation or mathematical operations can be applied. For example, the distances between different vectors can be used to approximate distances between words which can generalize the behavior from one word to another. These capabilities can be used in chatbots and virtual assistants as well who will be able to consider the topic of the question and the possible answers only with neural networks. The usage of vectors as the representation of words is part of the networks training process [12].

Recurrent Neural Networks (RNNs) are together forming a great and excessive model family, especially for sequence-based tasks. They are easily usable and practical because they have a high-dimensional hidden state with specific nonlinear dynamics that is enabling them to "remember" and furthermore process past information. In different problems and use-cases where the sequence is more important than all of the individual items, RNNs tend to shine.

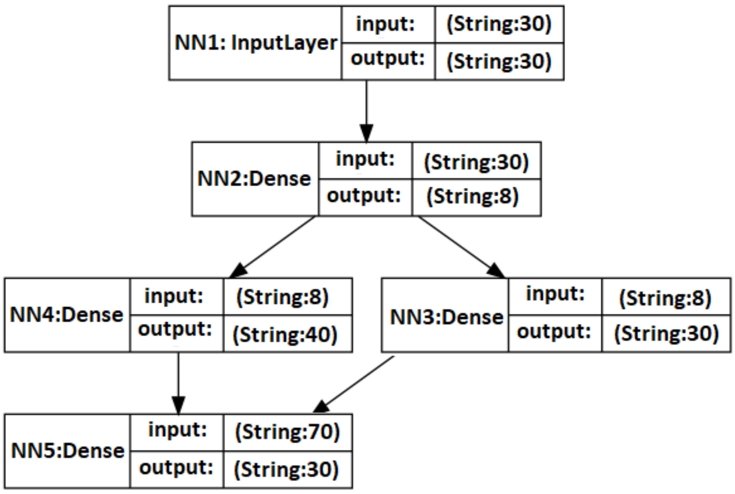

Fig. 1. Example of neural network combination diagram

We are basically talking about fully connected neural networks that can concatenate two different inputs in their hidden layers as well as doing another mathematical operation dependent on the specific activation function, we choose. RNNs are effective at solving time series problems as well but are worse in case tabular based data [13]. All these above-mentioned methods were considered in your application design and are proposed by us.

3 Methodology

In the very beginning, we need to define our functional as well as non-functional requirements. For the functional requirements we can propose the following:

- Ability to select the right topic of interest
- Ability to accept student question
- Ability to select the correct answer
- Ability to learn based on previous interactions

As non-functional requirements we can define the following:

- Accessible via the internet
- Usage of Python programming language

The basic functional requirement for the said system is the ability to integrate said assistant into any virtual learning environment (VLE) or learning management system (LMS). Therefore, it is necessary to provide a platform-independent solution, accessible via the Internet. This can only be achieved by creating a generally available application program interface (API) using the JSON format as a communication protocol.

The main task of this article is to design replicable design at the application level. From our point of view, this assistant can be divided into 3 basic components:

- User communication interface
- Neural network selector
- Neural networks

Said communication interface will serve as a tool for the student to communicate with the said assistant. Nowadays, it is considered common for relevant communication to take place via chat. An additional tool that can increase the attractiveness of this assistant is also the communication via an audio platform. This could be done by using speech-to-text conversion through the Google API and text-to-speech [14] using a service such as ResponsiveVoice. These services provide a wide range of interpretive languages, including the languages used in Central Europe.

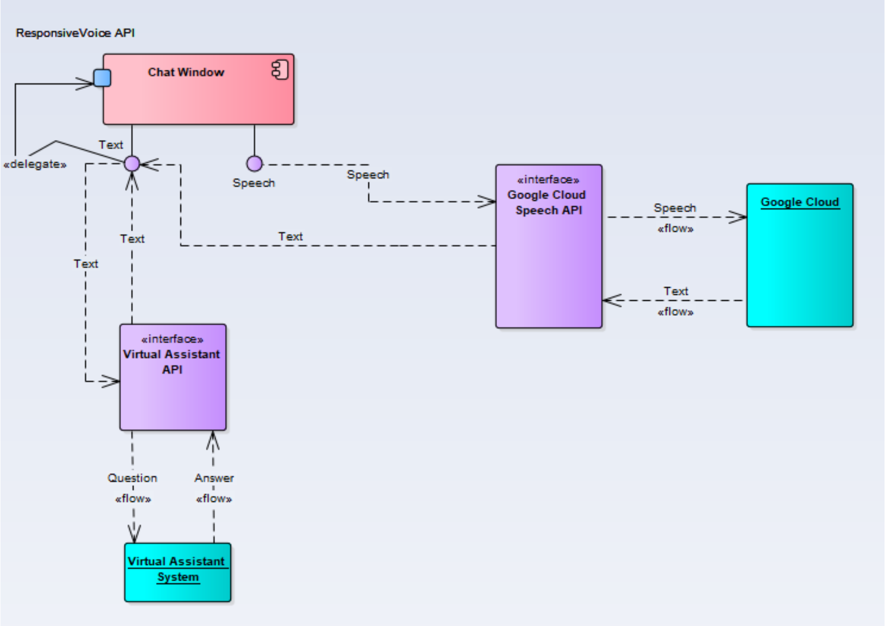

Fig. 2. Example of user communication interface

Figure 2 shows an example of the architectural design of such a communication interface. However, the communication interface itself is designed by the user of said system.

Based on our previous research we needed a rigorous methodology which can define final software architecture that will be used for implementation of our virtual assistant. We decided not to use any methodology from Quality Attributes category [21] but vice versa Analytic Hierarchy Process (AHP). AHP is a mathematical decision-making technique that proposed by Satty [22]. The AHP deals with problems of how to measure intangible criteria and how to interpret correctly measurements of tangibles, so they can be combined with those of intangibles to yield sensible, not arbitrary numerical results [23]. It is a widely used theory and provides a measurement through pairwise comparisons and relies on the judgments of experts to derive priority scales [24]. In order to apply AHP in an organized way to generate priorities; it needs to break down the decision into a few steps:

- Define the problems and determine the related knowledge
- Structure the decision hierarchy model from the top with the goal of the decision through the intermediate levels to the lowest level (a set of the alternatives)
- Construct a set of pairwise comparison matrices
- Weigh the priorities for every element by using the priorities obtained from the comparisons

The mathematics of the AHP and the calculation techniques are briefly explained in the following. Initially, assigning a number to each element on a scale that indicates how many times more important one element is over another element. The rating scale adapted from Saatys' fundamental scale of absolute numbers. These pairwise comparisons are carried out for all factors to be considered; after that, the matrix is completed. The next step is to calculate a consistency ratio (CR) to measure how consistent comparisons are. If the CR is less than 0.1, that indicates good consistency. The third step is to calculate the list of elements' priority vectors, which express the relative weight of each element type. The final stage is to compute the total score by adding the score of elements' priority values and the results with the highest total score is chosen [25].

Based on the presented methodology we created specialized framework which helped us decided which software architecture is the best for presented implementation. Result of this calculations was micro-servis architecture, mainly because of usage multiple independent neural networks.

A much more important part is the neural network selector. Its role can be considered critical to the practical success of this solution. Given the use of micro-service architecture, with the basic building block being a set of specialized neural networks, it is necessary to be able to identify which neural network should answer the relevant request (question) from the student. This selection is made by a standardized classifier. However, the problem occurred during the empirical verification of the correctness of the solution. We assumed that the most suitable tool for this task would again be a neural network that will act as a classifier. However, in the experimental verification, we found that we have similar tools at our disposal in terms of reliability, which are, however, much more optimal in terms of the impact on overall performance. All of the selected solutions were taken from Python library Scikit-learn [15].

Table 1. Avg. classifier performance on different data volumes

Data Volume	Decision Tree	SVM	Supervised Neural Network
1MB	0.00789 sec	0.00813 sec	0.00826 sec
100MB	0.01987 sec	0.02189 sec	0.02394 sec
1GB	2.82479 sec	5.04832 sec	3.64780 sec
5GB	3.90547 sec	7.31287 sec	4.52437 sec

The script which we used for performance experimentation has been executed on the same hardware and software equipment. The script has done the following common tasks:

- Import libraries
- Connect to MySQL 8.0 database
- Select all required data
- Create and train model
- Output result

The data set was a set of comments from one Slovak portal focused on international politics. Monitored time excluded all additional operations like connect to database etc. We focused only on creating a model, training, and outputting the result for our request. The classifier should decide what category (offensive, neutral, friendly) is most suitable for the provided sentence. We created 4 different data sets, so we can see what type of performance demand for each solution we can expect (linear, etc.). Based on our findings we decided that the best solution for this part of the proposed system will be the usage of DecisionTreeClassifier class from the mentioned library.

The last component of our virtual assistant system is the set of specialized neural networks. These networks are working out the answers to specific questions given to them from the previous categorization model. For this task, we decided to use an existing solution in the form of a library called TensorFlow 2 and more exactly a library built on top of it called Keras.

We chose Keras as our tool for RNN implementations because it is a widely used tool for RNN creation in a friendly easily understandable and scriptable manner. Our decision is supported by the guides on the Tensorflow documentation webpages, where the creators are proposing Keras as an easy to use and easy to customize way to work with specific layers of any RNN especially with multiple different layers of activation functions.

Keras provides a high number of suitable libraries to load the dataset, split the data, encode the data, and many more utility functionalities that we do not need to look for in different libraries. Furthermore, the different parts of the library support each other and are compatible which means that we do not need to do a high amount of data transformations to fit into the modeling functions of Keras during the training or testing of models. The outcome of the compatibility of utility functions comes with a potential speed boost during the script runtime.

A great selling point of Keras is the minimalistic configuration file which makes it easily configurable and it is supported with multiple virtualizations and containerization technologies like Docker [16]. Support of Docker will help us to achieve high availability of the application as well as the orchestration of different neural networks in our application microservice model.

Both our proposed activation functions are written into Keras and multiple optimizers are built-in as well. This will give us the capability to test different optimizers as well as activation layer functions for possible differences in our performance.

Our neural network of choice is a recurring neural network (RNN). It is a model specially designed for sequential data. Because sentences are from our point of view sequences of words, or from a perspective of spoken talk sequences of sounds they are a potentially good choice for the specific specialized answer creation. The recurrent neural networks have shown extraordinarily good performance in natural language processing types of tasks in the past [17].

As for the activation function, our chosen type is the LSTM (long-short term memory units) and GRU (gated recurrent unit). The reason for this choice is the loss of memory inside the RNN neuron. In the case of sequential data especially of sentences, the specific context might be crucial to decide what is the real point of the questions. This is moreover needed in case of young pupils who may ask questions in a very complicated way. Hence previous data points should be stored inside the memory of the RNN.

The global architecture of the described solution is represented in Figure 3. Based on the principles of software engineering, we can say that the design should meet the following characteristics:

- Scalability: The ability to add new components without significant intervention
- Accuracy: Thanks to the correct design, ensuring the highest possible accuracy of the answer
- High availability: The overall software should work even in the case of failed components
- Response speed: The ability to get responses in real-time.

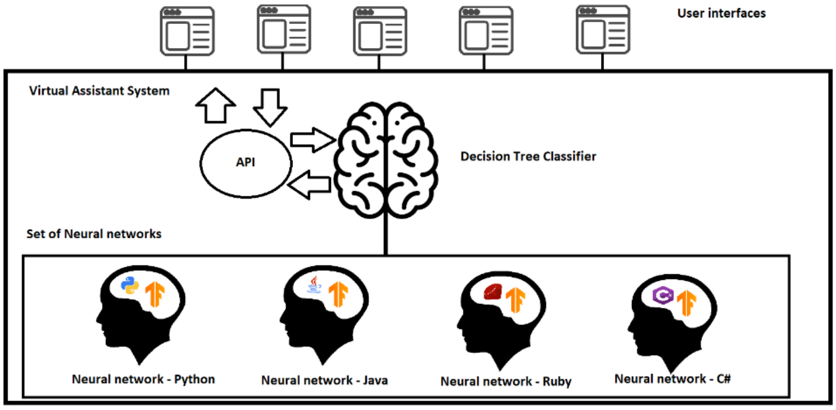

Fig. 3. Global architecture of virtual assistant system

At the same time, it must be said that the scenario that the neural network will not be able to answer should be included in the above concept, and therefore even if the classifier chooses it correctly, its content will not have information that would correctly answer the student's question. At that point, the system behaves by taking the 3 most likely answers and asking the student if one of them is correct. At the same time, it provides information about the probability to the student, so that the student is aware of the fact that the assistant is not sure of his answer. During this time, the system will automatically send the specified conversation to the system administrator for review so that he can complete the specified answer.

We introduce a deep neural network trained via interaction with students that provides a self-reflection possibility during training. These capabilities are allowing the RNN to validate its answer on more confusing or in some cases just difficult examples while still improving performance on the samples with high answer accuracy. We show that such self-reflection neural network can:

- Learn the representations for structured noise
- Enable more robust learning while getting arbitrary or unstructured noise by learning on noisy samples
- Be used as an effective detector for out of category samples that lean to reliably validate the output data when presented with samples from multiple unlearned topics

The main benefit of this structure is the ability to restructure the set of existing neural networks to the required shape. For example, based on the previous interactions we can see that the neural network created for answering questions for students searching for help in learning Java has been profiled for students with medium or even advanced programming skills. That is a problem when a student on a beginner’s level is trying to get information from it. What we can do is to create a new neural network that will be trained based on the interactions logged by the classifier until the point when the answer level becomes too complex. After this, we have 2 neural networks both of which are focused on 1 topic, but we can offer fewer complex answers based on the student level. Hence, we will simply add a new attribute to our classifier so it will take into account also this specific fact.

4 Discussion and Conclusion

Based on our analysis we can say, that the provided design is the best option to achieve a given goal. The system set up in this way brings a large number of advantages. The basic is the scalability of this solution. If necessary, we can add other knowledge units in the form of a neural network at any time so that they can respond to the needs of students. At the same time, this model is very accurate in choosing the answer due to the fact that at this level the system separates the knowledge sets and thus prevents irrelevant answers. A great advantage of such a chosen concept is the ability to tune the appropriate solution at the level of selection of a specialized neural network that is responsible for the response.

From a software orchestration perspective, the easiness to use Docker and virtualize the environments with the specific model with Keras running in the background will allow us to create networking communication between a high number of neural networks. This process is furthermore easy and manageable, because of orchestrators like Docker Swarm or Kubernetes. The application would be containerized and would be able to run even in the case of failed events or erroneous behavior, because the orchestrator is capable to restart any failed process without the need of human interaction. Rolling updates in the case of model retraining would be easy to set up and hence we do not need to be afraid of long down times during the model update processes.

As is well known, if we chose a neural network as a classifier, we would most likely achieve better results in terms of time response with a large amount of data, but the classifier would represent what a "black box" is. This would not allow us to detect and debug a poorly chosen neural network in a certain response. Therefore, we decided to choose decision trees that provide us with this advantage [18]. As a further advantage it can be considered on par with the neural network algorithm hence our execution and decision speed will not be penalized that much.

One of the challenges we need to consider is the possibility to train the model during production setting. In cases where the specific answer given by the network is not right, we need to create a solution how to report the specific issue. A possible solution is to allow the users especially the teachers to choose if the answer was right or wrong. In case that they do not consider it right they can give a proposal what is the right answer, and this message would be sent by email to the model creators. They can then allow or deny the request for change which would retrain the model with the new insight added into the dataset. Another solution is to make the model retrain itself in a short period of times where the new data would be considered. This solution without the middleman would make it possible to retrain the specific model with less issues and manual work. However, the negative aspect of this solution is that it can retrain the model in a wrong way or to retrain the wrong neural network overall in case that the topic decider model would send it to the wrong specialized neural network.

The biggest challenge for this system will be its correct training. There are several ways we can accomplish this task. Due to the fact that our virtual assistant is focused on assisting students in the Slovak Republic, it is not possible for us to choose existing data sets. Despite our efforts, we were unable to find a satisfactory data set that would be in the Slovak language and would contain any communication sessions related to IT topics. That's why we have more "raw" ways. An example of such a way of training said neural networks is crawling dictionaries and textbooks so that they can interpret the acquired text as knowledge for our neural network. However, if you choose this approach, we risk importing large amounts of unusable information that will reduce the overall accuracy as well as the performance of our virtual assistant. Therefore, we do not prefer this option as the primary solution to our problem.

We would consider the use of techniques known as Natural Language Generation (NLG) to be a suitable solution. It can be considered as a software process that automatically transforms different information into English or other language content. The technology is capable to tell a story by writing different sentences and/or texts for you. The problem with this approach is a lack of case studies done for Slovak language.

We also considered multiple open-source options for implementing presented solution. Based on research already carried out [26] we decided to choose a platform-independent solution in the form of a JavaScript plugin designed for variable use across existing standardized solutions as well as custom solutions of individual educational institutions.

5 References

[1] D. Halvoník, J. Kapusta, “Framework for E-Learning Materials Optimization”, in International Journal of Emerging Technologies in Learning, vol 15, issue 11 ISSN 1868-8799, 2020, p. 67-77.

[2] S. Thomas, E. Tewell, G. Willson, “Where Students Start and What They Do When They Get Stuck: A Qualitative Inquiry into Academic Information-Seeking and Help-Seeking Practices”, in The Journal of Academic Librarianship, vol 43, issue 3, ISSN 0099-1333, 2017, p. 224-231. https://doi.org/10.1016/j.acalib.2017.02.016

[3] X. Meng, G. Karniadakis, “A composite neural network that learns from multi-fidelity data: Application to function approximation and inverse PDE problems”, in Journal of Computational Physics, vol 401, ISSN 0021-9991, 2020. https://doi.org/10.1016/j.jcp.2019.109020

[4] S. Kim, J. Eun, Ch. Oh, B. Suh, J. Lee, “Bot in the Bunch: Facilitating Group Chat Discussion by Improving Efficiency and Participation with a Chatbot”, in CHI '20: Proceedings of the 2020 CHI Conference on Human Factors in Compu-ting Systems, ISBN 978-1-4503-6708-0, 2020, p. 1 - 13. https://doi.org/10.1145/3313831.3376785

[5] S. Arora, V. Athavale, H. Maggu, A. Agarwal, “Artificial Intelligence and Virtual Assistant—Working Model “, in Mobile Radio Communications and 5G Net-works, vol 140. Springer, ISBN 978-981-15-7130-5, 2020, p. 163 - 171. https://doi.org/10.1007/978-981-15-7130-5_12

[6] Y. Goldberg, Neural Network Methods for Natural Language Processing, 2017.

[7] T. Szabó and I. Pšenáková, "Computer-aided education and evaluation of student exam tasks from MS Excel," in 2017 15th International Conference on Emerging eLearning Technologies and Applications (ICETA), Košice, 2017. https://doi.org/10.1109/iceta.2017.8102536

[8] K. Cooper, S. Brownell, “Student Anxiety and Fear of Negative Evaluation in Active Learning Science Classrooms “, in Active Learning in College Science, Springer, ISBN 978-3-030-33600-4, 2020, p. 909 - 925. https://doi.org/10.1007/978-3-030-33600-4_56

[9] A. Hidayat, D. Suwawi, K. Laksitowening, “Learning Content Recommendations on Personalized Learning Environment Using Collaborative Filtering Method”, in 8th International Conference on Information and Communication Technology (ICoICT), Yogyakarta, Indonesia, ISBN 978-1-7281-6142-6, 2020, p. 1-6. https://doi.org/10.1109/icoict49345.2020.9166371

[10] T. Huyghebaert, N. Gillet, N. Beltou, F. Tellier, E. Fouquereau, “Effects of workload on teachers' functioning: A moderated mediation model including sleeping problems and overcommitment “, in Stress & Health, vol. 34, issue 5, ISSN:1532-2998, 2018, p. 589 - 680. https://doi.org/10.1002/smi.2820

[11] A. Androutsopouloua, N. Karacapilidisb, E. Loukisa, Y. Charalabidisa, “Transforming the communication between citizens and government through AI-guided chatbots”, in Government Information Quarterly, vol. 36, issue 2, ISSN: 0740-624X, 2019, p. 358-367. https://doi.org/10.1016/j.giq.2018.10.001

[12] I. Pšenáková, "Interactive Applications in the Work of Teachers," in XXIX. DIDMATTECH, Budapest, 2016.

[13] I. Sutskever, J. Martens and G. Hinton, "Generating Text with Recurrent Neural Networks," in ICML 2011, 2011.

[14] V. Këpuska, G. Bohouta, “Comparing Speech Recognition Systems (Microsoft API, Google API And CMU Sphinx)”, in Int. Journal of Engineering Re-search and Application, vol. 7, issue 3, ISSN: 2248-9622, 2017, p. 20 - 24. https://doi.org/10.9790/9622-0703022024

[15] S. Raschka, V. Mirjalili, "Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2, 3rd Edition", ISBN: 978-1789955750, 2019. https://doi.org/10.1002/9781119557500.ch5

[16] A. Gulli and S. Pal, Deep Learning with Keras, Packt Publishing Ltd, 2017.

[17] K. Yao, G. Zweig, M.-Y. Hwang, Y. Shi and D. Yu, "Recurrent Neural Net-works for Language Understanding," 2013.

[18] J. Kapusta, J. Obonya, “Improvement of Misleading and Fake News Classification for Flective Languages by Morphological Group Analysis”, in Informatics, vol 7, issue 1 ISSN 2227-9709, 2020. https://doi.org/10.3390/informatics7010004

[19] T. Szabo, R. Zitný, P. Ildikó, Z. Illés, and B. Viktória, “Using Mobile Technologies in University Education,” Nov. 2016, doi: 10.1109/ICETA.2016.7802042.

[20] Z. Illés, B. Viktória, T. Szabo, R. Zitný, and P. Ildikó, “Introducing Mobile Motivated Lectures,” Nov. 2016, https://doi.org/10.1109/iceta.2016.7802089

[21] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerland, M. Stal, “Pattern-Oriented Software Architecture”, in A System of Patterns, vol 1, ISBN 978-0-471-95869-7, 1996.

[22] T. Saaty, “The Analytic Hierarchy Process: Decision Making in Complex Environments “, in Quantitative Assessment in Arms Control, ISBN 978-1-4613-2805-6, 1984, p. 285-308. https://doi.org/10.1007/978-1-4613-2805-6_12

[23] T. Saaty, “Making and validating complex decisions with the AHP/ANP “, in Journal of Systems Science and Systems Engineering, vol 14, issue 1, ISSN 1861-9576, 2005. https://doi.org/10.1007/s11518-006-0179-6

[24] T. Saaty, “Decision making with the Analytic Hierarchy Process “, in International Journal of Services Sciences, vol 1, issue 1, ISSN 1753-1454, 2008, p. 83-98. https://doi.org/10.1504/ijssci.2008.017590

[25] M. Galster, A. Eberlein, M. Moussavi, “Systematic selection of software architecture styles “, in IET Software, vol 4, issue 5, ISSN 1751-8806, 2010, p. 349-360. https://doi.org/10.1049/iet-sen.2009.0004

[26] S. Papadakis, M. Kalogiannakis, E. Sifaki, N. Vidakis, “Evaluating Moodle use via Smart Mobile Phones. A case study in a Greek University”, in EAI Endorsed Transactions on

Creative Technologies, vol 5, issue 16, ISSN 2409-9708, 2018. https://doi.org/10.4108/eai.10-4-2018.156382

6 Acknowledgement

This work was supported by the Cultural and educational grant agency MŠVVaŠ SR under the contract No. KEGA-029UKF-4/2018.

7 Authors

Dominik Halvoník is with Department of Informatics Constantine the Philosopher University in Nitra, Nitra, Slovakia.

Peter Psenak is with Department of Quantitative Methods, Comenius University, Bratislava, Slovakia. psenak9@uniba.sk

Article submitted 2020-11-15. Resubmitted 2021-01-31. Final acceptance 2021-02-22. Final version published as submitted by the authors.