DOI: 10.51790/2712-9942-2020-1-2-4

THE APPLICATION OF MACHINE LEARNING AND NEURAL NETWORKS TO AUTOMATED TEXT AND VISUAL ASSIGNMENT VERIFICATION USED AS ASSISTANCE TO EDUCATORS

Nikita O. Besshaposhnikov[1,a], **Mikhail S. Diachenko**[1,b], **Alexander G. Leonov**[1,2,3,4,c], **Maxim A. Matyushin**[1,d], **Anton E. Orlovskii**[1,e]

[1] *Federal State Institution "Scientific Research Institute for System Analysis of the Russian Academy of Sciences", Moscow, Russian Federation*

[2] *Lomonosov Moscow State University, Moscow, Russian Federation*

[3] *Moscow Pedagogical State University, Moscow, Russian Federation*

[4] *State University of Management, Moscow, Russian Federation*

[a] *ORCID: http://orcid.org/0000-0002-7616-3143, nbesshaposhnikov@vip.niisi.ru*

[b] *ORCID: http://orcid.org/0000-0002-5809-4981, mdyachenko@niisi.ru*

[c] *ORCID: http://orcid.org/0000-0001-9622-1526, dr.l@vip.niisi.ru*

[d] *ORCID: http://orcid.org/0000-0003-1775-6894, itsaprank@yandex.ru*

[e] *ORCID: http://orcid.org/0000-0003-3823-0832, orlovskiy@niisi.ru*

Abstract: the digitalization of education in Russia and worldwide enables a more extensive introduction of advanced teaching methods through a partial switch from offline to online teaching. The existing and coming e-learning platforms feature not only digital lecture videos and e-textbooks but some automated assessment/grading tools. There is a need to expand the coverage of such tools to avoid the extreme burden of online teaching as the educator has to allocate significant time for assessing the increased amount of high school/university student assignments. Also, distant learning diminishes the effect of the educator personal presence since the teacher and the student are separated by their computer screens. Smart educator assistants and automated assessment tools based on machine learning and neural networks can significantly alleviate the problem. This study offers some strategies for automated assessment of graphic assignments and checks for plagiarism. Possible AI-based implementations of such features are presented.

Keywords: e-learning platform, checking for plagiarism, distant learning, automated assessment, smart educator assistant.

Acknowledgements: this study is supported by RFBR, project No. 19-29-14057.

Cite this article: Besshaposhnikov N. O., Diachenko M. S., Leonov A. G., Matyushin M. A., Orlovskii A. E. The Application of Machine Learning and Neural Networks to Automated Text and Visual Assignment Verification Used as Assistance to Educators. *Russian Journal of Cybernetics*. 2020;1(2):35–41. DOI: 10.51790/2712-9942-2020-1-2-4.

ИСПОЛЬЗОВАНИЕ МАШИННОГО ОБУЧЕНИЯ И НЕЙРОННЫХ СЕТЕЙ ДЛЯ АВТОМАТИЧЕСКОЙ ВЕРИФИКАЦИИ ЗАДАНИЙ В ТЕКСТОВОМ И ГРАФИЧЕСКОМ ПРЕДСТАВЛЕНИИ И ПОМОЩИ ПРЕПОДАВАТЕЛЮ

Н. О. Бесшапошников[1,а], **М. С. Дьяченко**[1,б], **А. Г. Леонов**[1,2,3,4,в], **М. А. Матюшин**[1,г], **А. Е. Орловский**[1,д]

[1] *Федеральное государственное учреждение «Федеральный научный центр Научно-исследовательский институт системных исследований Российской академии наук», г. Москва, Российская Федерация*

[2] *Московский государственный университет имени М. В. Ломоносова, г. Москва, Российская Федерация*

[3] *Московский педагогический госуниверситет, г. Москва, Российская Федерация*

[4] *Государственный университет управления, г. Москва, Российская Федерация*

[а] *ORCID: http://orcid.org/0000-0002-7616-3143, nbesshaposhnikov@vip.niisi.ru*

[б] *ORCID: http://orcid.org/0000-0002-5809-4981, mdyachenko@niisi.ru*

[в] *ORCID: http://orcid.org/0000-0001-9622-1526, dr.l@vip.niisi.ru*

[г] *ORCID: http://orcid.org/0000-0003-1775-6894, itsaprank@yandex.ru*

[д] *ORCID: http://orcid.org/0000-0003-3823-0832, orlovskiy@niisi.ru*

Аннотация: процесс цифровизации образования, активно проводимый в нашей стране и по всему миру, позволил более широко применить в учебном процессе современные приемы преподавания, перенося часть педагогической нагрузки с очного формата на дистанционный. Проектируемые и используемые цифровые образовательные платформы уже сейчас включают в себя не только оцифрованный лекционный видеоматериал и электронные формы учебников, но и элементы автоматизации проверки выполненных учащимися заданий. Расширение области применения автоматической проверки решенных учащимися задач и выполненных упражнений является объективной необходимостью, в противном случае при дистанционных формах образовательного процесса резко возрастает нагрузка на педагога, который должен выделять значительное время на проверку увеличившегося самостоятельной работы школьников и студентов. Кроме того, при дистанционном преподавании снижается эффект личного присутствия педагога, когда учитель и ученики разделены экранами компьютеров. Существенной помощью может стать использование интеллектуальных помощников преподавателя и автоматизированных систем проверки, построенных методами машинного обучения и технологии нейронных сетей. В настоящей статье рассмотрены подходы к решению поставленных задач по автоматической проверке графических заданий и выявлению заимствований в текстовом виде. Показаны возможные варианты реализации этих функций с использованием технологий искусственного интеллекта.

Ключевые слова: цифровая образовательная платформа, поиск заимствований, дистанционное образование, автоматическая проверка, интеллектуальный помощник преподавателя.

Благодарности: работа выполнена при финансовой поддержке РФФИ научного проекта № 19-29-14057.

Для цитирования: Бесшапошников Н. О., Дьяченко М. С., Леонов А. Г., Матюшин М. А., Орловский А. Е. Использование машинного обучения и нейронных сетей для автоматической верификации заданий в текстовом и графическом представлении и помощи преподавателю. *Успехи кибернетики*. 2020;1(2):35–41. DOI: 10.51790/2712-9942-2020-1-2-4.

We began to talk about a fully automatic learning process without the participation of a human teacher back in the 1970s when the first intelligent learning systems appeared [1, 2, 3], but the actual successes in this direction today do not allow us to fully replace traditional training in the format of a full-time school, university and classroom. Currently, there is a gradual transition of education to the digital form, which allows for wide accessibility and high efficiency of the educational process [4].

Digitalization of education has made it possible to move some of the learning processes from a face-to-face classroom format to a remote one, which has opened up the possibility of automating some of the functions of the human teacher, allowing to maintain the quality of learning and to fully use all the possibilities of the digital educational platform. Such functions, for example, include checking graphical assignments, estimating the probability of borrowing the solution of an assignment, including textual ones (anti-plagiarism), and the function of helping students find answers to common questions arising during the solution of assignments. Even though these functions are not traditional features of “intelligent learning systems” [5], for their effective implementation the use of artificial intelligence technologies is justified. The article considers the requirements for these functions and describes possible implementations of these functions using artificial intelligence technologies.

The automation of task checking is a typical task of a digital educational platform, which is especially in demand in programming education. Further in the paper, we will consider automatic checking of graphic tasks. We will call “graphic task” the task, the result of which is estimated from the image. The task of checking graphic tasks occurs, for example, in such disciplines as engineering and computer graphics [6]. In the case of computer graphics, the task checking can not be limited to a pixel-by-pixel comparison with the original and should allow counting the tasks with insignificant admissible deviations.

A graphical task is considered to be correctly solved if the image synthesized by the student’s program is close to what was required by the conditions of the task. One of the predefined functions can be chosen as a measure of closeness of the provided and expected solution, for example, a pixel-by-pixel comparison or an intelligent comparison with a reference. In this case, the teacher is not required to create a specialized software tool to check the tasks or to describe the results of the task in any formal language.

The verification function can be implemented both by classical methods of comparison with the reference and using modern intelligent methods, for example, based on the application of ANN [7]. In the

case of the application of intelligent methods for setting the function of checking the result, it is enough to provide the system with one or more images containing valid answers. As the result of the comparison determines the probability of proximity of the provided image to the expected one, in case of low probability the task is transferred to the final manual check by a human teacher. In practice, an intelligent system for checking graphical assignments can be composed of an embedder, which translates images into vectors of low dimensionality, and a KNN classifier, working in the space of these vectors (figure 1).

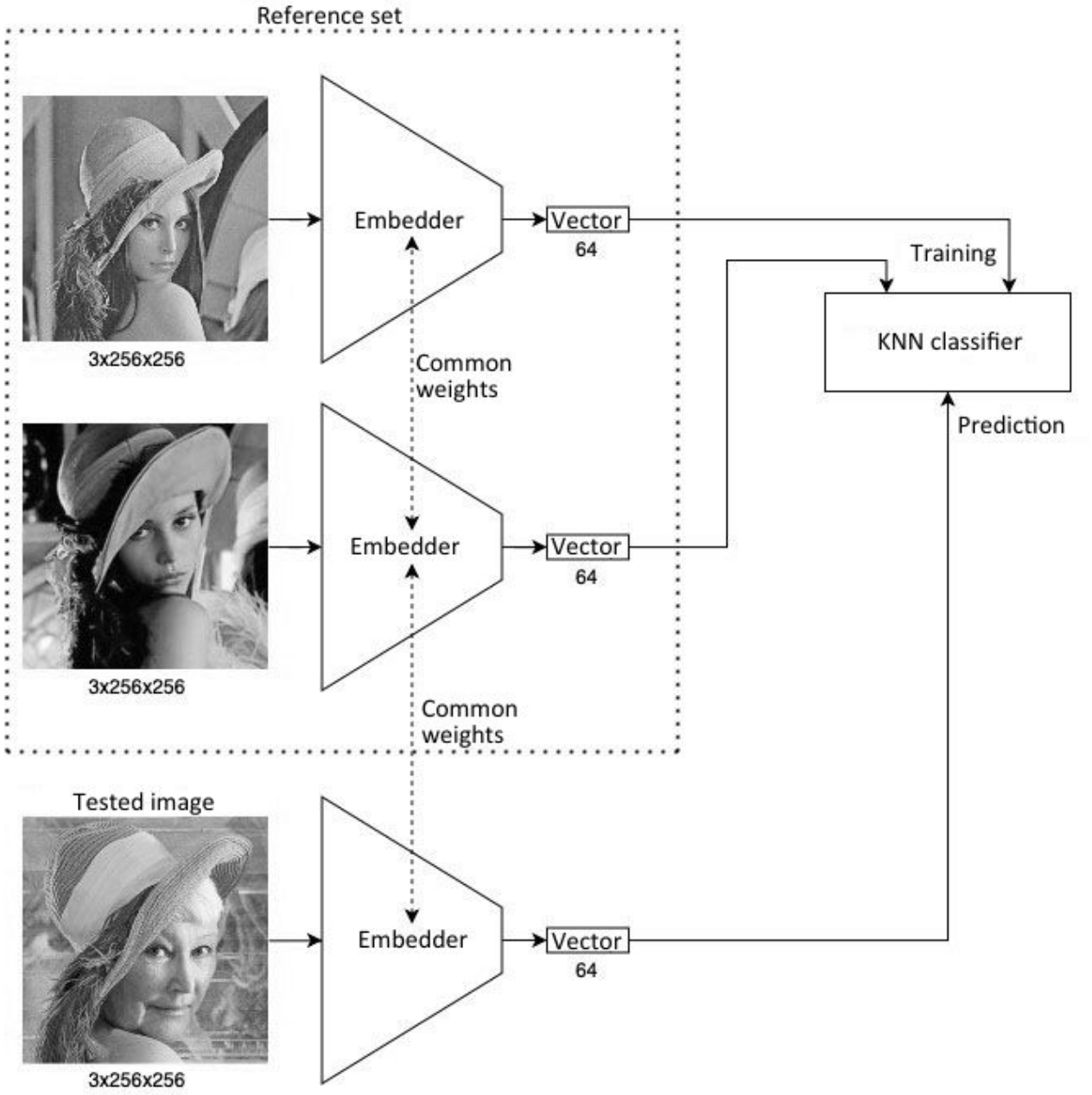

Figure 1. *Image comparator architecture with an embedder and a KNN classifier*

The embedder can be represented by an ANN, which can be trained on any available volumetric data set. During training the embedder is augmented by an extender to a full ANN of the following form: the embedder takes as input an image and outputs a low-dimensional vector, the extender takes as input this vector and outputs an image. During training, such an extender augmented ANN optimizes the pixel-by-pixel distance between the input and output images (Figure 2).

Thus, the Embedder is trained to encode high-dimensional spatial features embedded in the image into a low-dimensional vector. In this case, similar-looking images will be converted to vectors that are close in the Euclidean metric. It is noteworthy that it is not necessary to train the Embedder on new images to test the new type of problems, a single initially high-quality training on a sufficiently large set of pictures

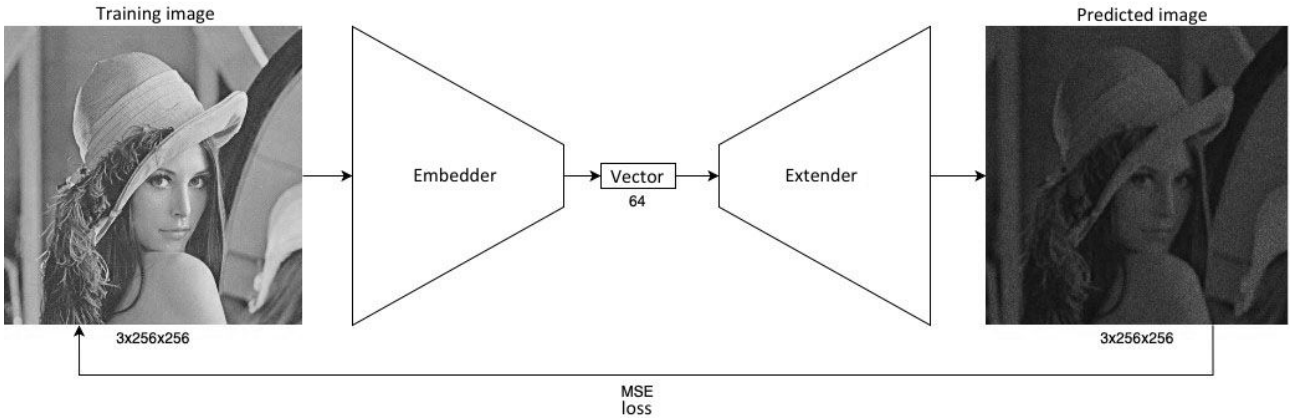

Figure 2. *Embedder training chart*

is enough.

For all the universality of intelligent methods in practice, the developers of verification systems resort to combining intelligent and heuristic methods, for example, to exclude the possibility of successful completion of the task when providing a blank, filled with one tone or containing a special "mask" (rather than the task solution) image.

In addition to graphical solutions, automatic verification can also be implemented in a wide range of tasks. For example, an obvious application is the automatic verification of solutions to programming problems, where it is sufficient to test the solution on a finite set of input data. And here artificial intelligence, oddly enough, can also be successfully applied, for example, to determine the so-called hardcode. Other cases, such as the automation of essay verification, without training and competent implementation of heavyweight neural network language models, are not feasible at all.

An equally urgent task in a remote environment is tracking and preventing the borrowing of solutions.

When working in the classroom, the problem of borrowing is solved by the teacher's communication with the student in the process of solving the task, which allows you to assess his level of understanding of the program provided and the degree of his "independence" in the preparation of the solution.

In the case of distance learning, there is no such possibility, so the teacher has to look through all solutions of tasks and analyze the possibility of borrowing them to maintain the quality of learning. The number of such tasks for each student can number in the dozens during the course. Let's consider this problem in the example of the task in the programming course.

The task of tracing borrowings is closest to the existing concept of "anti-plagiarism", that is, we are talking about the fact that, for example, when solving a programming problem, one student's solution was fully or partially used by one or more other students. A distinctive feature of educational programming tasks is that, firstly, the solution of simple (typical) tasks can be repeated with exact values of parameters by several students, and, secondly, the main part of the program can be a "template" of the program which is not changed by the student. Thus, only a small fragment of the program remains for the evaluation of borrowing. In this regard, there is no unambiguous solution to the problem of plagiarism assessment in this case, and it is only possible to indicate that several submitted solutions are similar, which is a signal to the teacher, who, drawing attention to the anomaly, will take action to prevent further copying of solutions, if it is confirmed.

Several techniques can be used to implement this function: style analysis, program structure analysis, problem-solving analysis, and heuristic analysis.

Analysis of style, in our case source code [8], is a kind of handwriting expertise. The limitation of this method's application is those novice developers either have not yet formed the style, or it repeats the industry standard, or the code can be automatically formatted by the development environment used. Signs of time can also include extensive copying fragments of solutions from programming sites or popular forums.

Methods of style analysis are implemented using ANN [8] and have proved to be good when analyzing large fragments of source code: the proximity of code styles of different students is evaluated and, if "suspicious proximity" is noticed, the task is sent to the teacher for checking. To implement such

methods, it is necessary to form a set of code metrics, based on which the comparison will be performed. Examples of metrics are indentation type, statements and block elements, naming character and length of names of parameters, functions and classes, etc.

The ANN itself is an embedder already familiar with recognizing solutions to graphical problems (Figure 3). The difference lies in the training process. Embedder for the task of plagiarism detection in practice is better trained as a Siamese network, using, for example, triplet loss.

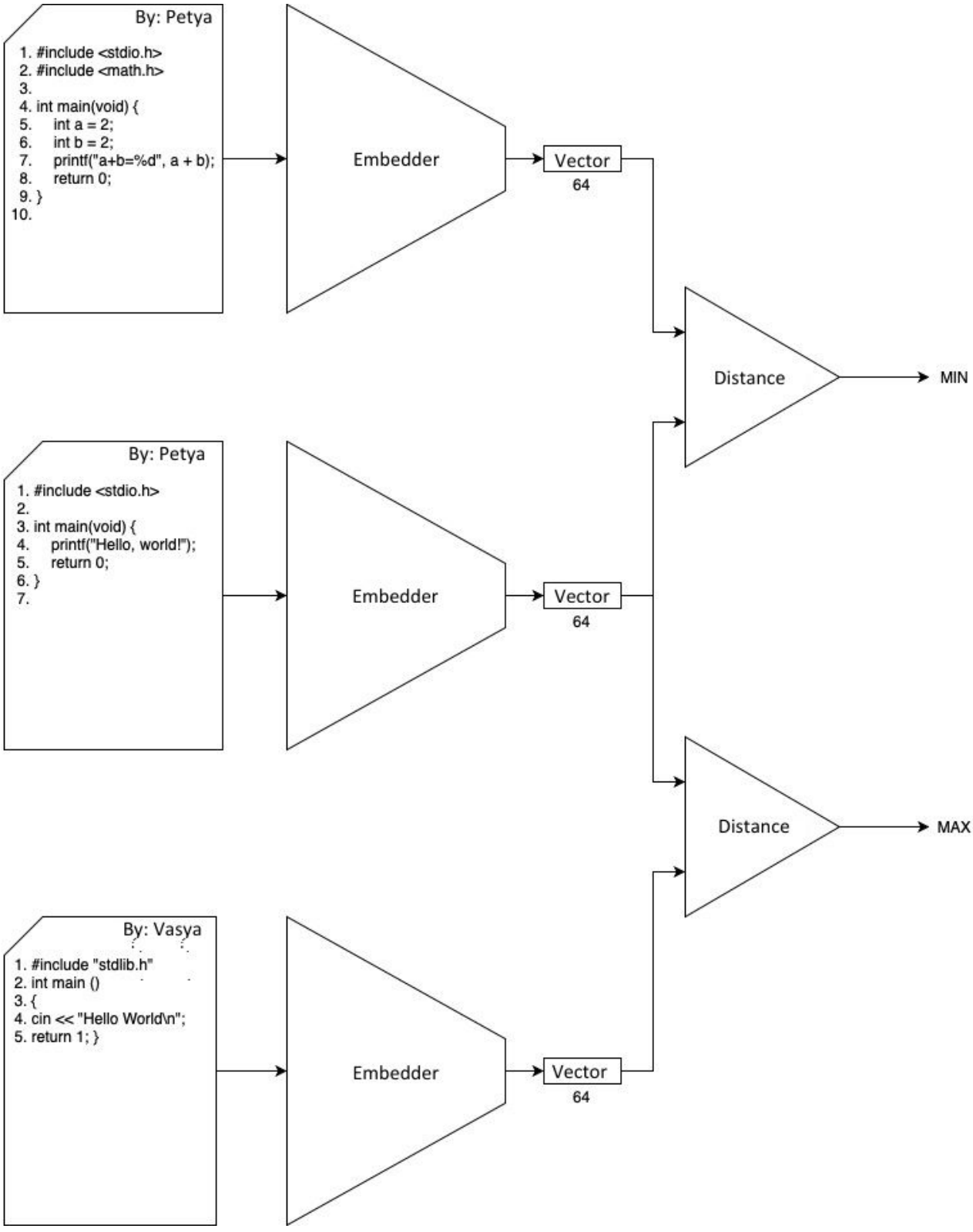

Figure 3. *The architecture of the system of source code comparison based on embedder*

Various open-source data sources can be used for learning, such as public repositories from GitHub. Examples of the same class in this approach are program codes belonging to the same author.

Program structure analysis can also be used for borrowing analysis. In this case, the program is analyzed and transformed into a representation that allows comparing the structures of two programs and their proximity [8]. Advanced intelligent implementations of such methods are resistant to the use of "obfuscation" – the addition of redundant language and algorithmic constructions to the code to reduce the "similarity" of two programs. For simple tasks, such methods will not show reliable results, because the structure of programs will be the same for all students.

The method of analyzing the problem-solving process is promising. In practice, even simple problems are not handed in on the first try. Thus, it is possible to analyze the evolution of the task to assess how the student progressed to the final solution. If the digital platform supports online editing of the assignment code, the analysis can be extended by using data on the student's work in the online editor. In this case, the problem-solution collects metrics about the nature of the code work, the specifics of entering language elements (parameter names, keywords), general typing parameters, etc. The application of ANN, in this case, will require the preparation of a large amount of data for training, which is a significant limitation for the application of this method at the stage of launching a new digital system.

Heuristic methods of analysis also go in combination with intelligent methods to rule out the simplest methods of borrowing, based on surrendering copies without change, etc.

Of course, any of the above methods can be bypassed by students, but the main function of checking is to prevent borrowing, not to catch them red-handed. Knowing that the assignment will be checked for borrowing, the student may decide to spend more time preparing an independent solution to avoid unintended consequences. The technology of generating unique assignments for each student can also be used to solve the borrowing problem, but this topic is the subject of a separate study.

In a distance learning environment, the issue arises of communicating answers to typical questions to students. In the classroom, such information spreads naturally in the course of the class, when the teacher answers one student's question and all the other students unwittingly listen to the answer. But in the case of distance learning, typical questions will come repeatedly. To relieve this burden the teacher may need an intellectual assistant. Below we will consider the dialog assistant (chat-bot).

The dialog assistant is the result of the development of information retrieval systems that provide answers to questions posed in the form of natural language statements. Although the very idea of dialog mode of communication with the student appeared in the '70s [3], this technology has become widespread only in the last 10 years due to the emergence of new technologies and computing tools for effective implementation of natural language analysis techniques.

The requirements for such systems may include support for students' vocabulary (abbreviations, slang, etc.), resistance to spelling and syntactic errors, collection of statistics on the questions for further analysis, the ability to redirect the question to the teacher if the assistant could not answer it.

Natural language analysis and dialog systems are high-tech intelligent solutions that require specialized knowledge, so it is important to provide the teacher with the ability to use such solutions at a high level, describing only the subject area in the form of typical questions and answers with reference material.

One solution in this area is the open-source library Rasa [5]. Based on this library, intelligent solutions can be built that do not require knowledge of specialized languages for describing dialog interfaces. Thus, the teacher can foresee the typical questions of students in advance and include them in the scripts of the dialog assistant's answer. Moreover, it becomes possible to transfer the control of the conversation to the teacher in case of the absence of the necessary answer to automatically "train" the dialog assistant to answer more students' questions later on.

Digital educational platforms at the current stage can be seen not as a substitute for the human teacher, but as a teacher's tool to automate resource-intensive activities. Automation frees up teacher's time to work with students who need extra attention and to continuously improve teaching methods. The usual human-teacher activities often turn out to be creative tasks that require either special knowledge to automate them or the use of intelligent technologies which take up all the complexity of automation.

REFERENCES

1. Shute V., Towle B. Adaptive E-Learning. *Educational Psychologist*. 2003;38.

2. Paviotti G., Rossi P., Zarka D. *Intelligent Tutoring Systems: an Overview*. 2012.
3. Carbonell J. AI in CAI: An Artificial-Intelligence Approach to Computer-Assisted Instruction. *IEEE Transactions on Man-Machine Systems*. 1970;11(4):190–202.
4. Besshaposhnikov N. O., Leonov A. G., Prilipko A. A. Digitalization of education – new opportunities for controlling educational tracks. *Proceedings in Cybernetics*. 2018; 2:154–160. (In Russ.)
5. Alesheva L. N. Intelligent Training Systems. *Vestnik Universiteta*. 2018;1:149–155. DOI: 10.26425/1816-4277-2018-1-149-155. (In Russ.)
6. Boykov A. A. Constructive Geometrical Basis and the Method of Automated Assessment of Drawings for Learning System. *Vestnik of Kostroma State University Series: Pedagogy. Psychology. Sociokinetics*. 2016;22(1):163–167. (In Russ.)
7. Koch G. R., Zemel R., Salakhutdinov R. Siamese Neural Networks for One-Shot Image Recognition. *ICML Deep Learning Workshop*. 2015;2.
8. *Machine Learning in Static Analysis of Program Source.* Available at: https://habr.com/en/company/pvs-studio/blog/484202/.
9. *Rasa*. Available at: http://rasa.com.