Development of a Chatbot Using Machine Learning Algorithms to Automate Educational Processes

Dmitry Alekseev [1], Polina Shagalova [1] and Eleonora Sokolova [1]

[1] *NNSTU n.a. R. E. Alekseev, Minina str., 24 Nizhny Novgorod, 603950, Russia*

Abstract

The use of chatbots in educational processes is relevant, where point communication with each student on common issues is required. A chatbot with artificial intelligence has been developed to automate educational processes. The cross-platform Telegram messenger is used to interact with the user. To increase the efficiency of creating a dataset, a graphical application interface in Python has been developed. Using libraries for creating graphical interfaces based on the Qt5 platform allows you to quickly navigate the intents, requests, responses that are already in the dataset. At the stage of developing the model structure, various vectorizers with different parameters were tested. To determine the intentions of users, a machine learning model was developed and implemented. The accuracy of the classification of user requests after training the model was 97%. An additionally developed algorithm based on the Levenshtein distance increased the classification accuracy. If the user's intent is not defined, a "stub" is triggered: "I did not understand the meaning of your question. Please rephrase it." Besides, the chatbot implements voice message recognition. As a result of the chatbot's interaction with users, statistics on requests are collected and all events occurring in the program are recorded. All information is presented graphically. After authentication, the user gets access to all statistics and can send messages on behalf of the bot, so the teacher can give a detailed answer. The architecture of the chatbot model allows it to be used on datasets of any educational process.

Keywords

Chatbot, machine learning, natural language processing, graphical interfaces, educational process

1. Introduction

Currently, artificial intelligence systems are developing, where one of the directions in the development of machine programs, chatbots that have artificial intelligence and interact with many users. The use of these technologies has great potential in the field of education, for example, in processes where the teacher spends a lot of time consulting students on typical issues. Chatbot development technologies are used to develop assistants for entering a higher educational institution, in online courses, in organizing students ' time, etc. [1]. At the same time, chatbots perform, as a rule, elementary functions, reducing time and routine work.

The paper presents a chatbot with artificial intelligence, trained on the developed dataset, which allows automating the process of passing the norm control by students-answers questions, sends the necessary documents to fill out, can connect a teacher to send more answers-consultations, provides statistics, and registers all events that occur. The chatbot selects the answer based on a given list of possible answers, using ranking technology.

GraphiCon 2021: 31st International Conference on Computer Graphics and Vision, September 27-30, 2021, Nizhny Novgorod, Russia
EMAIL: ada4667@yandex.ru (D. Alekseev); polli-shagalova@yandex.ru (P. Shagalova); essokolowa@gmail.com (E. Sokolova)
ORCID: 0000-0001-7826-483X (D. Alekseev); 0000-0002-6676-4228 (P. Shagalova); 0000-0003-0860-2463 (E. Sokolova)


2. Creating a dataset

For the process of passing the norm control, a dataset has been developed that represents a set of intentions of users – intents. Each intent includes examples of questions that users can ask, and the chatbot's answers to the questions asked (Figure 1). The implementation of the dataset includes 18 intents, the chatbot's questions, and answers were compiled on the basis of regulatory documents of the NNSTU n.a. R.E. Alekseev [2][3][4] for the implementation of the WRC. Each intent includes up to 20-30 questions.

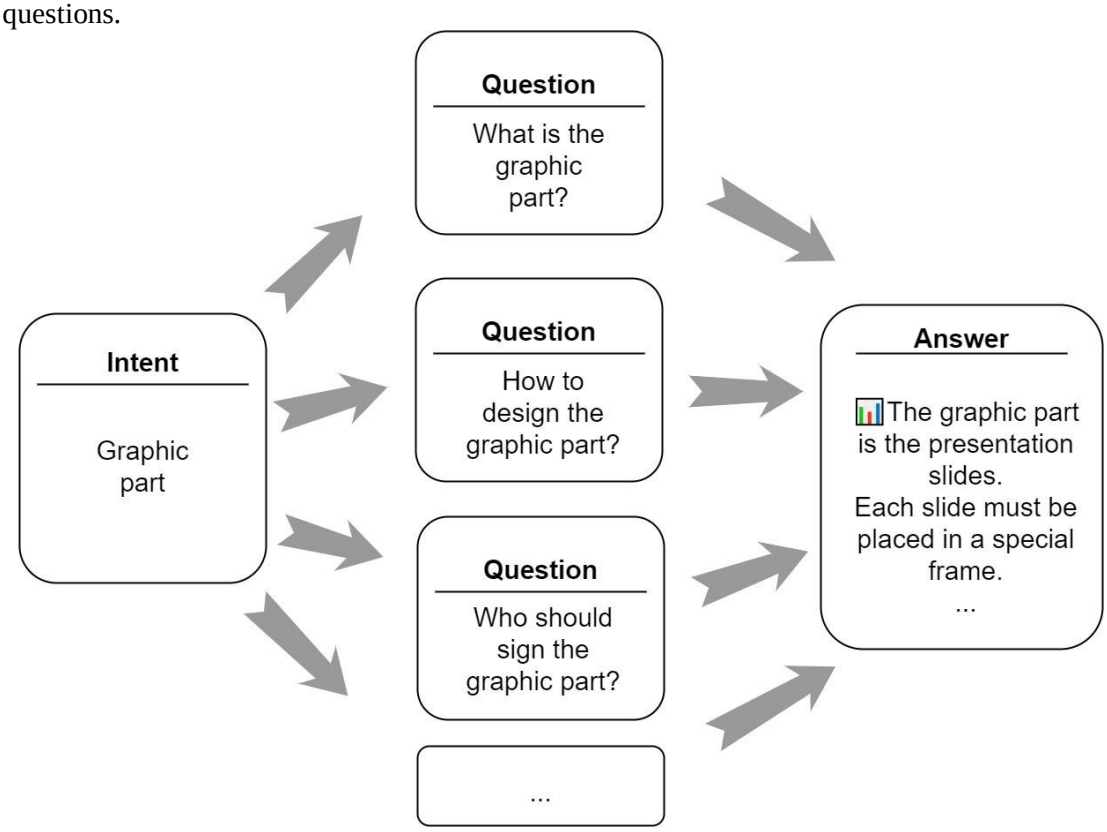

Figure 1: Example of an intent

Experience in the field of natural language processing has shown that it is inconvenient and inefficient to navigate a dataset and fill it out in a text editor. To increase the efficiency of work, a special application was created – a text editor. Its graphical interface allows you to quickly navigate in intents, requests and responses, that are already in the dataset (Figure 2).

To create the application, PyQt5 was used – a set of Python libraries for creating graphical interfaces based on the Qt5 platform [5]. The created dataset was used for training the model and the operation of an algorithm based on the Levenshtein distance.

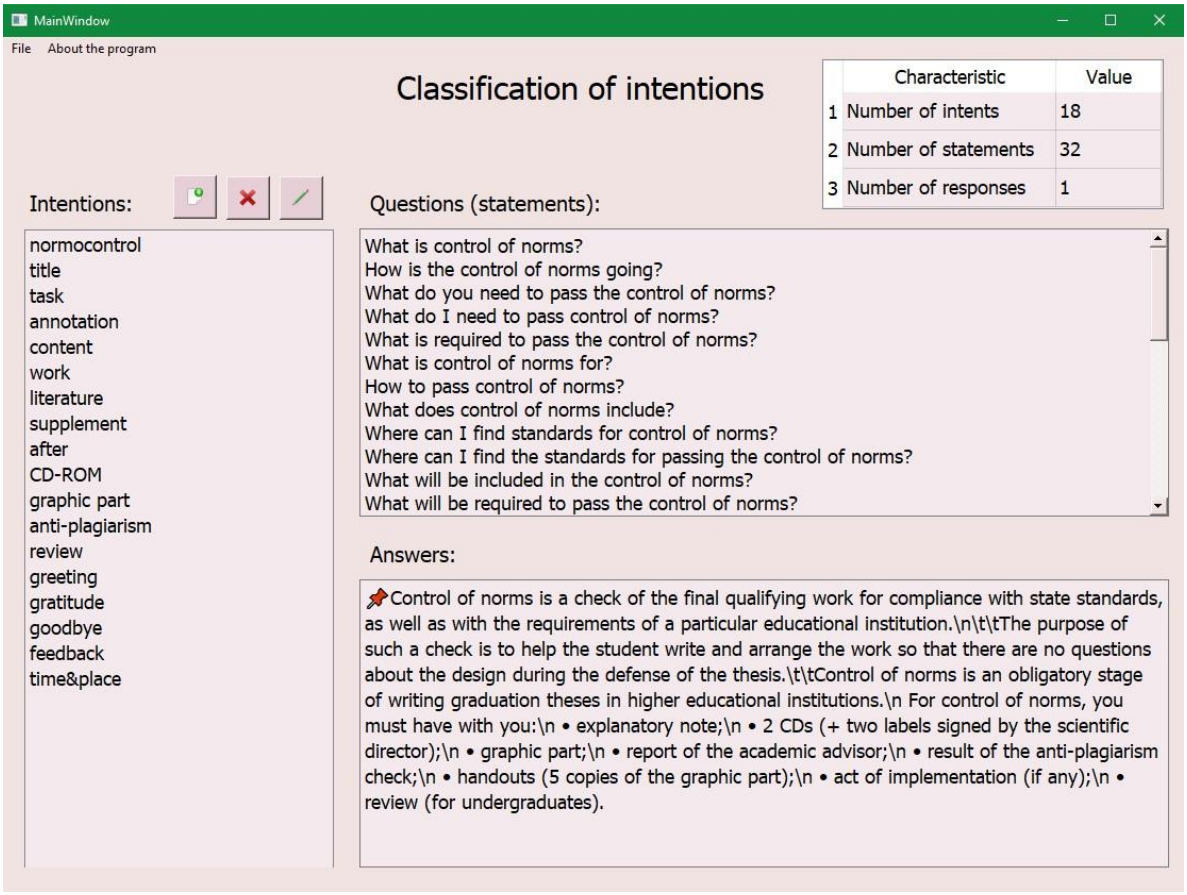

Figure 2: Application for parsing a dataset

3. Developing a machine learning model

To determine the intentions of users, a machine learning model has been developed that classifies user messages. Here, the class is the intent from the dataset. The stages of the classification algorithm are shown in Figure 3.

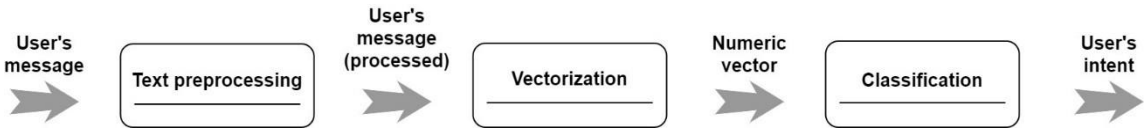

Figure 3: Classification algorithm

3.1 Text Preprocessing

Text preprocessing is a mandatory step in solving the problem of natural language processing, which allows increasing the accuracy of the classification of user intentions. It includes reducing words to lowercase, lemmatization, removing noise (non-letter characters), correcting grammatical errors in user queries.

In order for the machine learning model to perceive the same words written using different registers as the same user's intention, all words are reduced to lower case in the developed model. For lemmatization, the pymorphy2 library was chosen, the use of which showed the best results for a small amount of data at the input. As a result of lemmatization, word forms are reduced to a normal (dictionary) form [6] for their subsequent analysis. Noise removal consists in removing non-letter

characters – numbers, punctuation marks, extra spaces, special characters, or, for example, html tags. Removing noise and reducing words to lowercase are implemented using the string library. To correct grammatical errors in user requests, the pyaspeller library is used, whose tools, in case of an incorrect word (grammatical error), replace it with the closest correct form of the word.

3.2 Vectorization

Vectorization is the process of converting text into a numeric vector. To create a chatbot, an analysis of existing algorithms for creating vector representations of texts was performed, the following vectorization algorithms were selected and investigated [7]: CountVectorizer, TfidfVectorizer, HashingVectorizer. As a result of the research, the optimal values of the vectorizer parameters are found, presented in Table 1.

Table 1
Parameters of vectorizers

Vectorizer		Parameters of vectorizers
CountVectorizer	creating 2-3	the threshold for the frequency of ignoring terms when building a dictionary is 0.85 (if the threshold is exceeded, the terms are ignored)
TfidfVectorizer	character n-grams	- the threshold for the frequency of ignoring terms when building a dictionary is 0.85; - linear scaling
HashingVectorizer	to be extracted	not

3.3 Classification

To develop the classifier of intents, classical machine learning methods were considered [8]: the kNearest Neighbors method; the Decision Tree Classifier; the Naive Bayes method; Logistic Regression; the Support Vector Machines method.

All the parameters of the classification algorithms that were used in training are shown in Table 2.

For machine learning, the dataset was divided into 2 parts: 2/3 of the sample was used for training the model, 1/3 was used for testing. The data was divided into samples using stratification, which made it possible to increase the classification accuracy for intents with an unequal number of examples in the dataset. To select the optimal parameters of each classification algorithm, a cross-validation mechanism was used, implemented using GridSearch, a tool of the sklearn library for automatically selecting the best hyperparameters of the model in a fixed grid of possible values.

To assess the effectiveness of the classification algorithms, the following metrics were considered [9]: the proportion of correct answers of the classification algorithm (accuracy); accuracy (precision) – a number of intents that are really objects of this class among all intents assigned by the chatbot to this class; completeness (recall)-the proportion of found intents of the class among all intents of the class.

Table 2

Configuring classifier parameters

Classifier	Parameters	Initial value	Final value	Step
K-Nearest Neighbors	number of "neighbors"	2	10	1
Decision Tree Classifier	maximum tree depth	2	10	1
	a function for dividing data into subclasses	- entropy (the more homogeneous the set, the less entropy) - error of the 1st kind (the frequency of a randomly selected example of a training sample will be classified incorrectly, gini)		
Naive Bayes	a priori probabilities of classes	they are selected automatically		
Logistic Regression	regularization method	methods are investigated: Lasso regression, Ridge regression, Elastic-net		
	class weights	- balanced (inversely proportional to the frequencies of classes in the input data), - they are selected automatically		
	maximum number of training iterations	100		
Support Vector Machines	regularization parameter (selection of significant features)	1,0	2,0	0,25
	class weights	- balanced (inversely proportional to the frequencies of classes in the input data), - they are selected automatically		
	maximum number of training iterations	100		

3.4 The results obtained

Based on the data obtained, it is concluded that the best options for the vectorizer and classifier will be the CountVectorizer and the support vector method (Figure 4). The accuracy of the classification of user requests by the model was 97%.

Detailed classification report (test data):

	precision	recall	f1-score
CD-ROM	1.00	1.00	1.00
after	1.00	1.00	1.00
annotation	1.00	1.00	1.00
anti-plagiarism	1.00	1.00	1.00
content	1.00	1.00	1.00
feedback	1.00	1.00	1.00
goodbye	1.00	0.75	0.86
graphic part	1.00	1.00	1.00
gratitude	0.75	1.00	0.86
greeting	0.75	0.75	0.75
literature	1.00	1.00	1.00
normocontrol	1.00	1.00	1.00
review	1.00	1.00	1.00
supplement	1.00	1.00	1.00
task	1.00	1.00	1.00
time&place	1.00	1.00	1.00
title	1.00	1.00	1.00
work	1.00	1.00	1.00
accuracy			0.97

Figure 4: The result of training the model

4. An algorithm based on the Levenshtein distance

To improve the accuracy of intent recognition and reduce the number of situations when the chatbot will not be able to determine the user's intention and answer his question (in this case, a "stub" is triggered), a modified Levenshtein algorithm was developed and applied [10], based on the calculation of the "editorial distance" metric – the difference between two sequences of characters. The value of the Levenshtein metric is determined by the minimum number of operations of replacing, inserting, deleting one character when converting one string (word) to another. This algorithm is included in the processing of intents if the machine learning model cannot classify the user's intention. In this case, the user's message is compared with messages from the dataset and the Levenshtein distance is calculated. The ratio distances of the Levenshtein to the length of the message from the dataset is taken as a configurable parameter of the modified algorithm:

$$\frac{distance(user\ request)}{length(example\ of\ a\ request\ from\ a\ dataset)} < 0{,}2, \quad (1)$$

where distance is the Levenshtein distance; length is a function that calculates the number of characters in a string.

The threshold value is empirically determined to be 0.2. If the ratio value is less than the threshold, then the user's intention coincides with the intention to which the example from the dataset is attributed.

5. Creating a Telegram chatbot

To interact with users, the free cross-platform messenger Telegram was used [11], which allows users to exchange text, voice messages, as well as media files of various formats. The Python-telegrambot library, created for the development of bots for Telegram, provides the ability to add various functionalities for bots, for example, sending messages, files, processing commands (a line starting with a slash character "/", including up to 32 characters of the Latin alphabet, numbers, and underscores), etc. [12].

The developed machine learning model and an algorithm based on the Levenshtein distance were integrated into a chatbot in Telegram. To start working with a chatbot, you need to type the name of the bot — Normobot in the Telegram search. When you enter the start command, a welcome message appears with an explanation of the chatbot's operation (Figure 5).

At the request of the user, the bot can send the necessary regulatory documents. So, in Figure 6, under the message from the bot with an explanation of what the norm control is, there are two buttons, "Regulation on the verification procedure", "Regulation by type of activity". When the button is clicked, the corresponding document will be sent to the user.

For the convenience of the user when communicating with the chatbot, voice message recognition has been added. The algorithm voices for processing messages includes, obtaining the id of a file with a voice message; downloading this files with the OGG extension; converting it to a file with the WAV extension; reading a file with a message, and converting it into a text form; transmitting a text message to a machine learning model. Converting a file from the OGG extension to the WAV extension is implemented using the pydub library. The audio recording is converted to text using the Speech Recognition library.

Figure 7 shows an example of a voice request. During the recording of the message, the question was asked, "what to do after passing the norm control?".

There are two buttons under the input line, "Checklist" and "Main errors". When you click on the "Checklist" button, the user will receive a list of documents necessary for passing the norm control procedure. After clicking the "Basic errors" button, the user will receive information about the mistakes that students make most often when preparing documents for standard control.

To work with the chatbot the administrator mode is implemented. After entering the password, the user gets additional features, such as viewing the program logs, viewing statistics on requests to the chatbot and sending messages on behalf of the chatbot (Figure 8). The added convenient function of sending messages on behalf of the chatbot allows the teacher to give an extended answer to the questions asked.

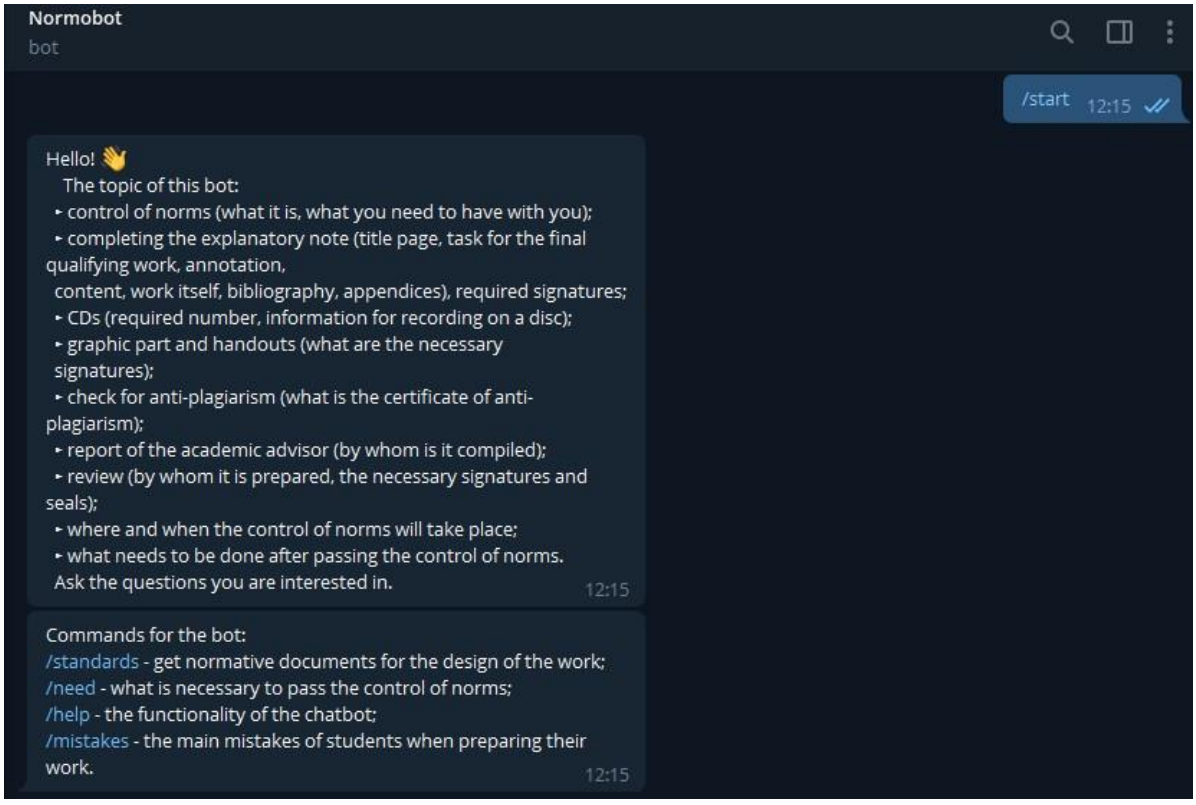

Figure 5: Getting started with a chatbot

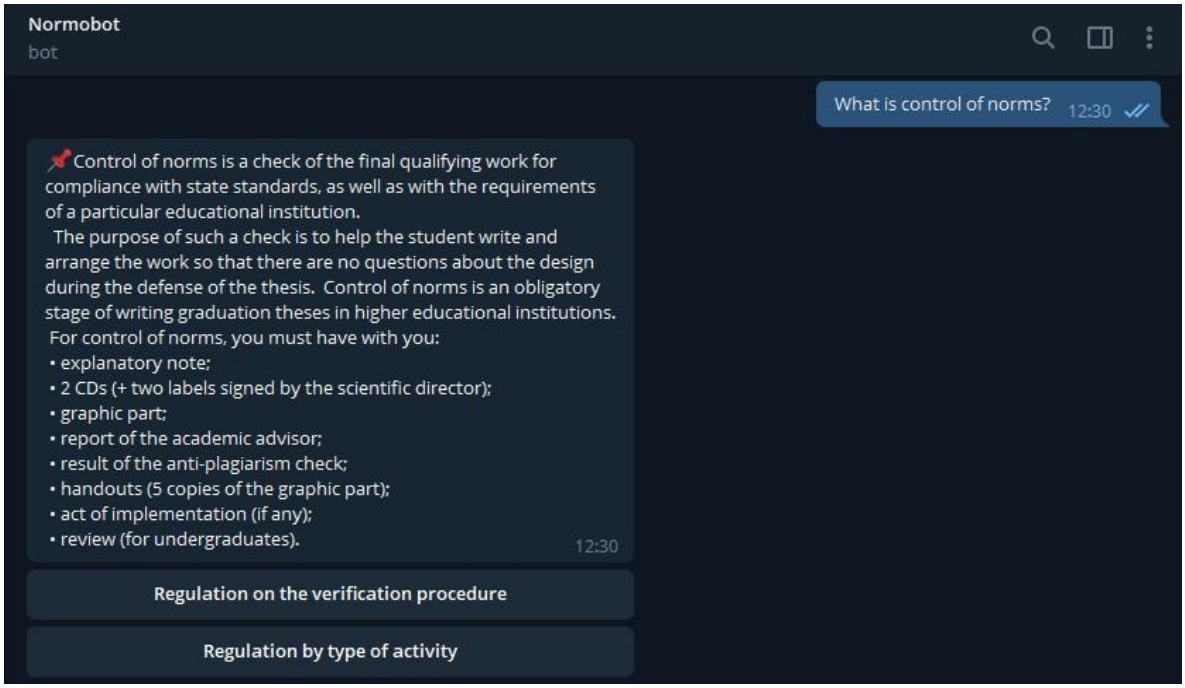

Figure 6: Getting started with a chatbot

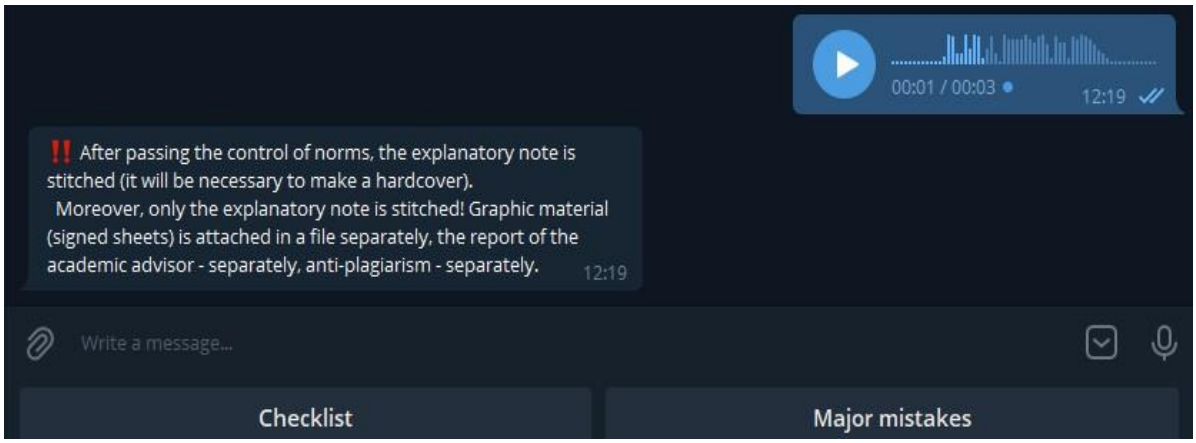

Figure 7: Example of a voice request

The Logging library is used for logging, i.e. writing data about the program's operation to a file on the disk, which is called a log or log. The logging data is displayed in the console and saved in a file (Figure 9).

Information about user requests is saved in a file with the CSV extension. This functionality was implemented using the built-in CSV library. A screenshot of the file with user requests is shown in Figure 10. The password required to enter the administrator mode is located in a separate file. This allows you to change the password without reassembling the project.

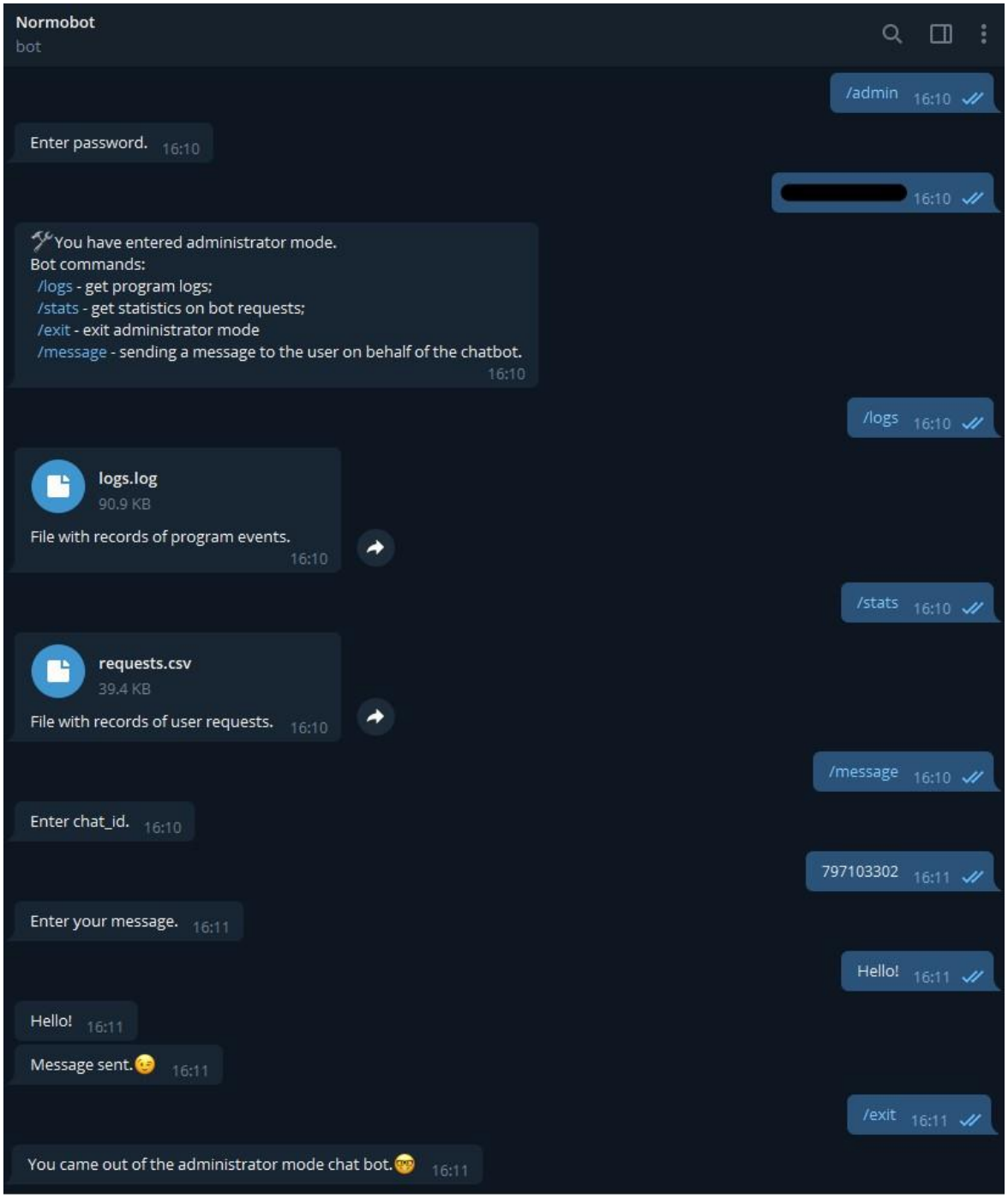

Figure 8: Administrator Mode

```
2021-05-12 14:55:36,305 - Normobot - INFO - Launch chat bot
2021-05-12 14:55:36,601 - Normobot - INFO - Bot info: {'id': 1318466039, 'first_name': 'Normobot', 'is_bot': True, 'usernam
2021-05-12 14:55:54,589 - Normobot - INFO - Command need
2021-05-12 14:55:55,935 - Normobot - INFO - Command mistakes
2021-05-12 14:56:23,235 - Normobot - INFO - Voice message processing
2021-05-12 14:56:24,886 - Normobot - INFO - Save statistics
2021-05-12 14:56:52,535 - Normobot - INFO - Attempt to log in to Admin mode
2021-05-12 14:57:08,157 - Normobot - INFO - Enter admin, chat_id:797103302
2021-05-12 14:57:13,164 - Normobot - INFO - Sending stats
2021-05-12 14:57:15,992 - Normobot - INFO - Exit admin, chat_id:797103302
```

Figure 9: Logging

	A	B	C	D	E	F	G
1	Number	Chat id	Date	Type message	Question	Answer	Method
2	1	797103302	2021-05-19 10:02:39+00:00	text	'Hi'	'Welcome!'	ML model
3	2	797103302	2021-05-19 10:02:44+00:00	text	'who prepares an anti-plagiarism certificate'	'Plagiarism is copied content published on anot	ML model
4	3	797103302	2021-05-19 10:02:50+00:00	audio	'what disks are needed'	'Two CDs must be provided for control of norms	ML model

Figure 10: Request file

6. Conclusion

As a result of the research, a chatbot with artificial intelligence has been developed to automate the process of standard control of the WRC, which is able to send documents to the student for standard control, give advice on the design of an explanatory note and other regulatory documents, check the correctness of the design of documents, and enable teachers to give advice to students on behalf of the bot. An extension of this project is the development of an algorithm for automatically filling out the documents necessary for passing the standard control and automating the process of checking the WRC for anti-plagiarism, as well as using it in other educational processes on the corresponding datasets.

7. References

[1] O. A. Yudin, I. A. Yudin, Writing a chatbot assistant for entering a higher educational institution. Modern science: Actual problems of theory and practice, Series: natural and technical sciences 6(2) (2019) 117–122.

[2] National standard GOST R 7.0.100–2018. Bibliographic record. Bibliographic description. General requirements and rules of compilation. Moscow: Standartinform, 2018. 128 p. URL: http://www.skunb.ru/data/upload/documents/files/ibo/GOST_new.pdfD0%A2_%D0%A0_7_0_100_2018_1204.pdf.

[3] NNSTU LDPE 11.2/34-18. Position by type of activity. About the final qualifying work on educational programs of higher education of NNSTU-Nizhny Novgorod: NNSTU named after R.E. Alekseev, 2018, 38 p. URL: https://www.nntu.ru/frontend/web/ngtu/files/org_structura/upravleniya/umu/docs/norm_docs_ngt u/pologenie_vipysk_rab_opop.pdf?23-04.

[4] NNSTU-LDPE-11.3-04-17. Regulations on the procedure for checking final qualifying works for the amount of borrowing and their placement in the electronic library system of NNSTU-Nizhny Novgorod: NNSTU named after R. E. Alekseev, 2017, 12 p. URL: https://www.nntu.ru/frontend/web/ngtu/files/org_structura/upravleniya/umu/docs/norm_docs_ngtu/polog_o_poryadke_proverki_vkr.pdf.

[5] PyQt5, Python Package Index, 2021. URL: https://pypi.org/project/PyQt5.

[6] I. Akhmetov, A. Krassovitsky, I. Ualiyeva, R. Mussabayev, A. Gelbukh, Lemmatization of russian language by tree regression models, Research in Computing Science 149(3) (2020) 147-153.

[7] 4 methods of text vectorization, Python School, 2020. URL: https://python-school.ru/nlpvectorization-methods/

[8] Overview of classification methods in machine Learning using Scikit-Learn, Tproger, 2019. URL: https://tproger.ru/translations/scikit-learn-in-python/

[9] Metrics in machine learning tasks, Habr, 2017. URL: https://habr.com/ru/company/ods/blog/328372/

[10] J. McConnell. Analysis of algorithms. Active learning approach: a textbook-Moscow: Technosphere, 2018, 416 p.

[11] 6 indisputable advantages of Telegram bots over mobile applications, sites and groups in social networks, Hab, 2015. URL: https://habr.com/ru/post/296388/

[12] Library in Python python-telegram-bot, DOCS Python3, 2021. URL: https://docspython.ru/packages/biblioteka-python-telegram-bot-python/.