

УДК 004.032.26

DOI: 10.18413/2518-1092-2024-9-4-0-6

Колесников В.Д.
Кабальянц П.С.

ПРИМЕНЕНИЕ МОДЕЛЕЙ РЕКУРРЕНТНЫХ НЕЙРОННЫХ СЕТЕЙ ДЛЯ ГЕНЕРАЦИИ ТЕКСТОВЫХ ДАННЫХ СПЕЦИФИКАЦИИ СБОРОЧНЫХ ЧЕРТЕЖЕЙ

Белгородский государственный технологический университет им. В.Г. Шухова,
ул. Костюкова, 46, Белгород, 308012, Россия

e-mail: kolesnikov_vladm@edu.bstu.ru, p.s.k@list.ru

Аннотация

В статье исследуются возможности различных моделей рекуррентных нейронных сетей в генерации текстовых данных. В частности, рассматриваются модели классической рекуррентной сети (RNN), сети с долгой краткосрочной памятью (LSTM) и генеративно-сопоставительной сети (GAN) в контексте задачи генерации текста спецификации для сборочных чертежей согласно формату, утверждённому ГОСТом. Для обучения моделей был использован набор данных на русском языке, расширенный дополнительными записями, имитирующими входные наборы деталей чертежей и ожидаемый текст спецификаций. Сделан вывод, что максимальное соотношение ответов без ошибок ко всем ответам, сгенерированным моделями, имеют генеративно-сопоставительные сети (GAN).

Ключевые слова: генерация текста; языковые модели; нейронные сети; рекуррентные нейронные сети; генеративно-сопоставительной сети; спецификация чертежей

Для цитирования: Колесников В.Д., Кабальянц П.С. Применение моделей рекуррентных нейронных сетей для генерации текстовых данных спецификации сборочных чертежей // Научный результат. Информационные технологии. – Т.9, №4, 2024. – С. 51-57. DOI: 10.18413/2518-1092-2024-9-4-0-6

Kolesnikov V.D.
Kabalyants P.S.

APPLICATION OF RECURRENT NEURAL NETWORK MODELS FOR GENERATING TEXT DATA OF SPECIFICATIONS OF ASSEMBLY DRAWINGS

Belgorod State Technological University named after V.G. Shukhov,
46 Kostyukova St., Belgorod, 308012, Russia

e-mail: kolesnikov_vladm@edu.bstu.ru, p.s.k@list.ru

Abstract

The article explores the capabilities of various models of recurrent neural networks in text data generation. Specifically, classic recurrent network (RNN), long short-term memory network (LSTM) and generative adversarial network (GAN) models are considered in the context of the problem of generating specification text for assembly drawings according to the format approved by government standard. To train the models, a data set in Russian language was used, expanded with additional records simulating input data, consisting of the drawing parts, and expected text of the specifications. The input data set for the study was divided into four groups of equal size, depending on three main factors: amount of input parts, their repeatability and grammatical complexity. It is concluded that for all four groups of input data generative adversarial networks (GANs) have the maximum ratio of error-free responses to all responses generated by model, followed by LSTMs and, lastly – RNNs. As a result, it is planned to use GAN-based models in future researches on specification text data generation.

Keywords: text generation; language models; neural networks; recurrent neural networks; generative adversarial networks; drawing specification

For citation: Kolesnikov V.D., Kabalyants P.S. Application of recurrent neural network models for generating text data of specifications of assembly drawings // Research result. Information technologies. – Т.9, №4, 2024. – P. 51-57. DOI: 10.18413/2518-1092-2024-9-4-0-6

ВВЕДЕНИЕ

Нейронные сети открывают большие возможности во многих сферах деятельности, так как они позволяют решать обширный спектр задач. Технологические достижения в области искусственного интеллекта дали возможность вычислительным машинам выполнять задачи высокого уровня, которые считались недостижимыми ранее.

Одним из крупных классов задач являются задачи генерации текста. Данный класс также в свою очередь включает в себя генерацию разных типов текстовых данных, таких как новостные статьи, стихотворения, сценарии и др. В зависимости от типа генерируемого текста варьируется настройка используемых алгоритмов и данных, на которых происходит обучение модели. Возникает вопрос о выборе наиболее подходящих языковых моделей для генерации конкретных текстовых данных, в зависимости от задачи.

Исследования показали, что крайне эффективными искусственными сетями для работы с большим количеством неструктурированных данных являются глубокие нейронные сети (англ. Deep Neural Networks, DNN) [2]. В виду того, что объёмные наборы текстовых данных, на которых происходит обучение нейронных, являются в большинстве своём неструктурированными, DNN хорошо подходят для выполнения задач по генерации текста.

Наиболее распространёнными рассматриваемыми моделями для генерации текста являются цепи Маркова, сети с долгой краткосрочной памятью (англ. Long Short-Term Memory, LSTM), рекуррентные нейронные сети (англ. Recurrent Neural Networks, RNN) и генеративно-сопоставительные сети (англ. Generative Adversarial Networks, GAN) [5, 7].

Результаты множества исследований показывают эффективность разных типов искусственных нейронных сетей для выполнения задач по генерации текстовых данных конкретного типа. В некоторых случаях наибольшая эффективность достигается в результате комбинирования нескольких моделей.

Например, по оценкам индивидуальных характеристик каждой модели было выявлено, что для генерации текстовых детских историй, среди нейронных сетей классов LSTM, RNN и GAN (в данном случае рассматривалась модель MaskGAN), наиболее высокие показатели имела генеративно-сопоставительная сеть, вслед за которой шла рекуррентная сеть, и лишь затем – модель с долгой краткосрочной памятью [3].

Другое исследование, в котором рассматривались цепи Маркова, LSTM и Модель с управляемыми рекуррентными блоками (англ. Gated Recurrent Units, GRU) в контексте задачи генерации текстов песен разных жанров, показало, что в среднем наибольший индекс читаемости имеют модели LSTM, а плотность рифмы – GRU [6].

Актуальными в последнее время также являются исследования возможностей генеративно-сопоставительных сетей в выполнении широкого класса задач по генерации текстовых данных. Концепция GAN, впервые представленная в 2014 году Иэном Гудфеллоу [8], первоначально рассматривалась в контексте генерации изображений, однако дальнейшие исследования показали эффективность применения данных моделей и для генерации текста [1, 4, 5].

Со временем появлялись новые подвиды моделей генеративно-сопоставительных сетей, такие как SeqGAN, LeakGAN, MaliGAN, SentiGAN, CatGAN и др. [1], полученные в результате усовершенствования классической модели GAN разными алгоритмами, что в свою очередь улучшало показатели данных моделей и качество генерируемого текста. Многие из данных моделей показывали эффективность в выполнении конкретных классов задач по генерации текстовых данных. Исследования также выявили, что некоторые ограничения технологии GAN, например невозможность дифференциации выборок из распределения по дискретным объектам, возможно избежать с применением распределения Gumbell-Softmax [9].

Также рассматривались комбинации рекуррентных нейронных сетей и генеративно-сопоставительных сетей с дополнительным применением технологии обучения с подкреплением, что в свою очередь показало высокую эффективность при генерации категориальных текстовых данных [4].

В целом, использование нейронных сетей для выполнения задач по генерации текста является актуальной на сегодняшний день проблемой, по которой ежегодно проводится большое количество исследований и разработок. Искусственный интеллект и генераторы текста применяются передовыми компаниями, такими как OpenAI (ChatGPT), Google (Gemini), Яндекс (YandexGPT) и др., и регулярно дорабатываются. Также разрабатываются языковые модели для более узких задач по генерации конкретных видов текстовых данных [3, 6].

В данной статье будут рассматриваться возможности различных моделей нейронных сетей в контексте выполнения задачи по генерации спецификации, по входящим данным сборочных чертежей с учётом формата, утверждённого государственным стандартом (ГОСТ).

ПОСТАНОВКА ЗАДАЧИ

Процесс оформления технических чертежей и их спецификаций является крайне трудоёмкой задачей, требующей соблюдения многочисленных правил оформления, описанных в ГОСТе. Использование нейронных сетей для генерации таких текстовых данных позволило бы значительно ускорить работу подобного плана.

Целью данной работы является изучение эффективности языковых моделей на основе RNN, LSTM и GAN в генерации спецификации к сборочным чертежам. Предполагается, что рассматриваемым генераторам на вход поступает информация о деталях, их количестве и размерах. Задача генератора – вернуть спецификацию, соблюдающую все требования ГОСТа и содержащую всю переданную информацию в нужном формате.

Критериями эффективности полученных результатов являются лексическая, орфографическая и синтаксическая корректность генерируемого текста и соблюдение правил ГОСТа для спецификации к чертежам. Пример спецификации для сборочного чертежа приведён на рисунке 1.

Формат	Зона	Поз	Обозначение	Наименование	Кол.	Примечание
				<u>Документация</u>		
A4			XX.XXX.00.00 СБ	Сборочный чертёж		
				<u>Детали</u>		
A4	1		XX.XXX.00.01	Корпус	1	
A4	2		XX.XXX.00.02	Шпindelь	1	
A4	3		XX.XXX.00.03	Гайка накидная	1	
A4	4		XX.XXX.00.04	Крышка	1	
A4	5		XX.XXX.00.05	Втулка сальниковая	1	
A4	6		XX.XXX.00.06	Кольцо поднабивочное	1	
A4	7		XX.XXX.00.07	Золотник	1	
A4	8		XX.XXX.00.08	Прокладка	1	
A4	9		XX.XXX.00.09	Прокладка	1	
A4	10		XX.XXX.00.10	Маховик	1	
				<u>Стандартные изделия</u>		
		11		Гайка М8 × 1,25 ГОСТ 5915-70	1	
		12		Шайба 8 ГОСТ 11371-78	1	
				<u>Материалы</u>		
		13		Шнур пеньковый	0,05 кг	
			XX.XXX.00.00			
Изм.	Лист	№ докум.	Подп.	Дата		
Разраб.					Лист	Листов
Проб.						1
И контр.						
Чтб.						
				Вентиль		

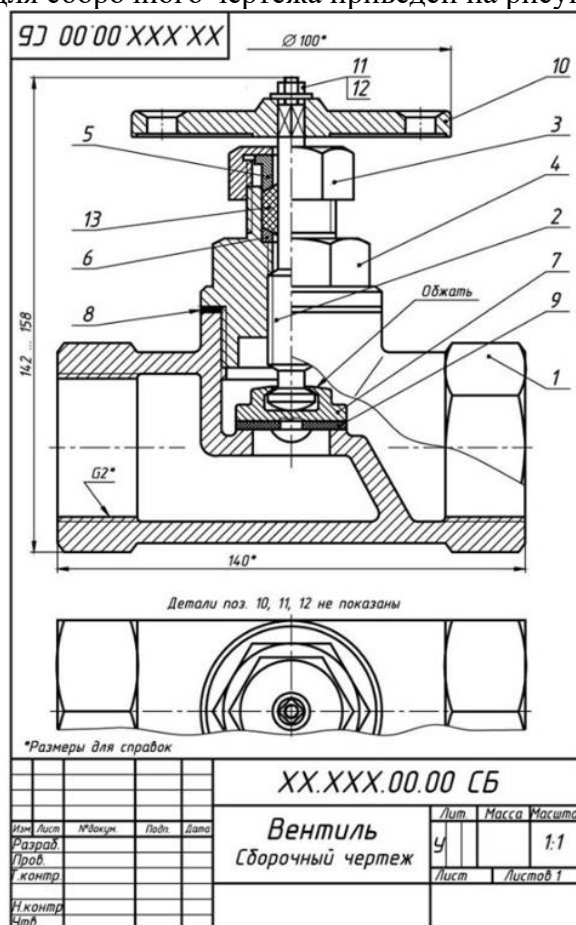


Рис. 1. Пример спецификации (слева) для сборочного чертежа вентиля (справа)
Fig. 1. Example specification (left) for valve assembly drawing (right)

Стоит учитывать, что в качестве ответа от генератора рассматривается текстовое содержание таблицы спецификации, непосредственно без оформления самой таблицы и основной надписи.

Процесс анализа моделей включает в себя несколько этапов. На первом этапе планируется выбор подходящих моделей для каждого класса нейронных сетей. Предполагается использование максимально базовых версий моделей, не включающих в себя никаких алгоритмов или надстроек, усложняющих работу нейронных сетей. Все модели будут обучены на одном и том же наборе данных, обучение каждой модели проходит с одинаковым количеством эпох. В качестве набора данных будет выступать «Открытая семантика русского языка».

Второй этап предполагает генерацию текстовых данных каждой моделью и сравнение сгенерированного текста с ожидаемым результатом. На данном этапе предполагается подобрать 50 наборов входящих данных и соответствующих им ожидаемых результатов. Для каждого набора данных генерация текста будет проводиться 100 раз. Результаты каждой генерации будут проверяться на корректность с учётом правил написания русского языка, и затем на полное или частичное соответствие ожидаемым результатам для входящего набора данных.

В качестве входных данных предполагается использовать список дателей, к сборочному чертежу. Например, для рисунка 1 входными данными является следующий текст: «корпус, шпindelь, гайка накидная, крышка, втулка сальниковая, кольцо поднабивочное, золотник, прокладка, прокладка, маховик»

В качестве выходных данных ожидается текст, содержащий непустые строки таблицы согласно ГОСТ в виде «Название столбца: Значение», разделённых запятой для каждого из изделий из списка входящих данных. Строки разделяются точкой с запятой. Таким образом, для данных рисунка 1 ожидаемый результат следующий: «Формат: А4, Позиция: 1, Обозначение: ХХ.ХХХ.00.01, Наименование: Корпус, Количество: 1; Формат: А4, Позиция: 2, Обозначение: ХХ.ХХХ.00.02, Наименование: Шпindelь, Количество: 1; Формат: А4, Позиция: 3, Обозначение: ХХ.ХХХ.00.03, Наименование: Гайка накидная, Количество: 1;...» и аналогично для оставшихся деталей.

Такой формат был выбран с целью упрощения дальнейшего заполнения таблицы спецификации сгенерированными данными, в случае подключения подходящего модуля.

ИССЛЕДОВАНИЕ И РЕЗУЛЬТАТЫ

Для генерации данных в конечном итоге были выбраны следующие модели нейронных сетей:

1. Для реализации RNN использовалась библиотека Keras. В частности, была выбрана трёхслойная модель на основе классов данной библиотеки. Входной слой формирует векторное представление данных (слой Embedding). Он кодирует входящие слова в n-разрядные числовые вектора. В качестве выходного слоя использовался полносвязный слой нейронной сети (слой Dense). Для скрытого внутреннего слоя применяется базовая реализация RNN от Keras.

2. LSTM была также реализована при помощи трёхслойной модели из библиотеки Keras. Входной и выходной слои аналогичны слоям модели RNN, а для внутреннего слоя используется базовая реализация LSTM.

3. Для модели GAN использовалась библиотека TextGAN-PyTorch. Из данной библиотеки была выбрана реализация генеративно-сопоставительной сети последовательностей с градиентом (SeqGAN) [10].

В набор данных для обучения нейронной сети также была добавлена 1000 произвольно сгенерированных пар входных данных и соответствующих ожидаемых ответов. Это позволило повысить качество получаемых в результате работы моделей текстовых данных по критерию ГОСТа. Обучение происходило в 20 эпох. Для этого использовалась ЭВМ со следующими параметрами:

- Процессор: AMD Ryzen 5 5500U 2.1 ГГц
- Видеокарта: Radeon Graphics
- Оперативная память: 8 Гб

Входные данные для исследования были разделены на 4 равных по количеству группы в зависимости от их сложности:

1. Примитивные данные – содержат наборы из 5-8 неповторяющихся деталей, названия которых состоят из максимально коротких и простых слов.

2. Простые данные – содержат наборы из 10-20 неповторяющихся деталей, названия которых состоят в основном из простых слов небольшой или средней длины (в большинстве своём не более десяти символов). Допустимы названия из нескольких коротких слов.

3. Данные средней сложности – содержат наборы из 15-25 различных деталей, которые также могут повторяться в входном списке. Названия могут включать в себя одно или несколько слов любой длины.

4. Данные высокой сложности – аналогичны наборам средней сложности, однако к ним добавляется ещё один список: стандартные изделия. Два списка разделены точкой. Пример итоговой структуры входящих данных: «Детали: гайка накидная, кольцо поднабивочное, втулка сальниковая. Изделия: гайка М8×1,25, ГОСТ 5915-70»

После обучения моделей и генерации текста по всем четырём группам заготовленных входных данных, были получены результаты, представленные в таблице и на рисунке 2.

Таблица

Результаты исследования эффективности генерации текста для спецификации

Table

Results of the study on the efficiency of text generation for specification

Сложность входных данных	Соотношение количества результатов с ошибками к общему количеству результатов, %					
	Грамматические ошибки и искажение названий			Несоответствие ожидаемой структуре ответа		
	LSTM	RNN	SeqGAN	LSTM	RNN	SeqGAN
Примитивные данные	3.46	3.54	3.43	17.21	17.87	16.69
Простые данные	4.61	4.85	4.49	20.31	21.62	19.19
Данные средней сложности	6.53	7.03	6.29	29.85	32.65	27.45
Данные высокой сложности	10.02	11.11	9.52	49.85	55.51	45.01

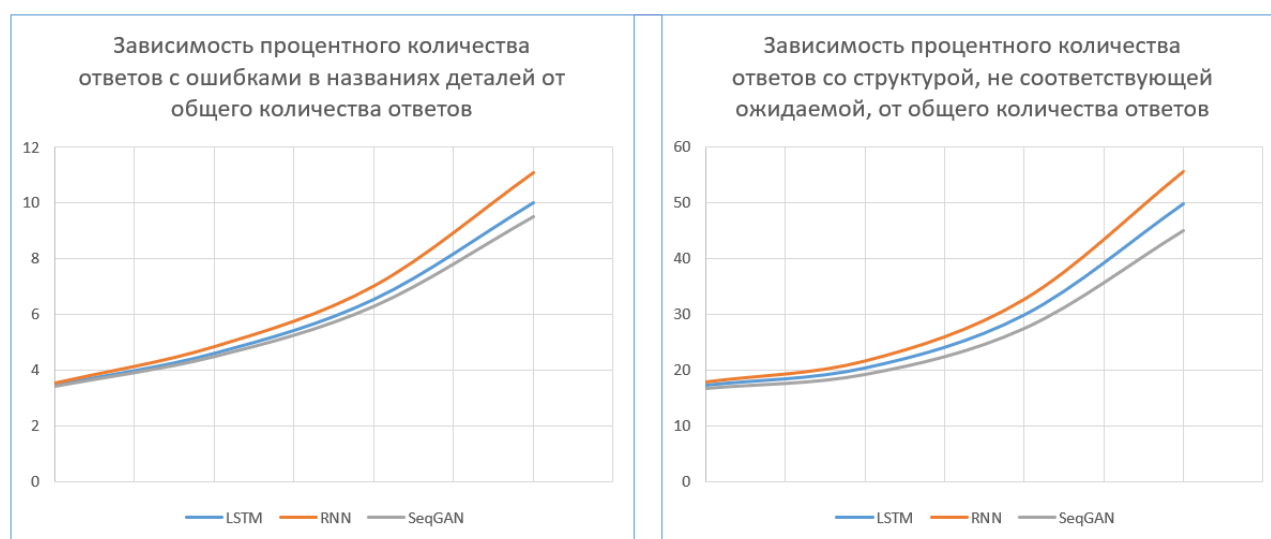


Рис. 2. Графики зависимостей количества ошибок для разных моделей
Fig. 2. Graphs of the dependencies of the number of errors for different models

По результатам можно сделать вывод, что наиболее эффективной для генерации требуемых текстовых данных оказались модель GAN, следом за ней – LSTM, и затем RNN. При этом стоит отметить, что эффективность генерации текста заметно снижается с увеличением сложности

входных данных. Высокие показатели несоответствия ожидаемой структуре ответа связаны, в первую очередь, с ограниченными ресурсами, задействованными при проведении исследования. Качество генерируемого текста напрямую зависит от количества данных, на которых обучаются модели; в данном исследовании размер набора данных был относительно небольшим. Также стоит учитывать возможности ЭВМ, на которой проводилось обучение моделей и последующая генерация текста.

ЗАКЛЮЧЕНИЕ

Результаты исследования показывают достаточно высокую эффективность GAN-моделей для генерации текстовых данных. Стоит заметить, что в данном случае использовались наиболее простые модели текстовых генераторов на основе GAN, RNN и LSTM, но даже такие модели смогли показать удовлетворительные показатели для решения поставленной задачи. Усовершенствование моделей с помощью дополнительных алгоритмов распределения (например, Gumbell-Softmax) и увеличение размера набора данных для обучения модели позволило бы добиться лучших результатов.

В дальнейшем планируется развивать модель GAN для более эффективной генерации качественных текстовых данных спецификации сборочных чертежей. Текущие наработки показали, что модель в целом способна справиться с требованиями ГОСТа и возвращать корректные выходные данные, хоть и не в 100% случаев.

Важно подметить, что ни одна из рассматриваемых моделей не была обучена или настроена специально для решения конкретной поставленной задачи. Для получения оптимальных результатов существенную роль играет настройка параметров модели; каждая узконаправленная задача генерации текстовых данных требует своей реализации и настройки модели генератора.

Список литературы

1. Imran A.S., Yang R., Kastrati Z., Daudpota S.M., Shaiks S. The impact of synthetic text generation for sentiment analysis using GAN based models // Egyptian Informatics Journal. – Volume 23. – Issue 3. – 2022. – pp. 547-557.
2. Subasi F. Practical Machine Learning for Data Analysis Using Python. Academic Press. – 2020. – pp. 91-202.
3. Shroufi D., Bayas A., Joshi N., Misal M., Mahajan S., Gite S. Story Generation Using GAN, RNN and LSTM. In: Garg D., Rodrigues J.J.P.C., Gupta S.K., Cheng X., Sarao P., Patel G.S. (eds) Advanced Computing. IACC 2023. Communications in Computer and Information Science. – vol. 2053. – 2024. https://doi.org/10.1007/978-3-031-56700-1_16, pp. 193-204.
4. Li Y., Pan Q., Wang S., Yang T., Cambria, E. A Generative Model for category text generation. Information Sciences. – 2018. – Volume 450. – pp. 301-315.
5. Dychka I., Legeza V., Oleshenko L., Bohutskiy D. Advances in Computer Science for Engineering and Education III, 2020. – pp. 344-346.
6. Tee T.H., Bei Yeap B.Q., Gan K.H., Tan T.P. Learning to Automatically Generating Genre-Specific Song Lyrics: A Comparative Study. In: Villazón-Terrazas B., Ortiz-Rodriguez F., Tiwari S., Sicilia M.A., Martín-Moncunill D. (eds) Knowledge Graphs and Semantic Web. KGSWC 2022. Communications in Computer and Information Science, vol 1686. https://doi.org/10.1007/978-3-031-21422-6_5, 2022. – pp. 62-75.
7. Dhall I., Vashisth S. Micro-Electronics and Telecommunication Engineering. 2020. – pp. 649-657.
8. Goodfellow I.J., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville F., Bengio Y. Generative Adversarial Nets. Département d'informatique et de recherche opérationnelle, Université de Montréal, Montréal, 2014. – p. 9.
9. Kusner M., Hernández-Lobato J.M. GANS for Sequences of Discrete Elements with the Gumbel-softmax Distribution, 2016. – p. 6.
10. Yu L., Zhang W., Wang J., Yu Y. SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient, Shanghai Jiao Tong University, 2016. – pp. 2852-2858.

11. Зувев С.В., Кабальянц П.С., Поляков В.М. Выявление аномалий в потоке с помощью фрактальной размерности графа нейронной сети обработки данных // Информационные системы и технологии. – 2021. – № 5 (127). – С. 31-38

References

1. Imran A.S., Yang R., Kastrati Z., Daudpota S.M., Shaiks S. The impact of synthetic text generation for sentiment analysis using GAN based models // Egyptian Informatics Journal. – Volume 23. – Issue 3. – 2022. – pp. 547-557.
2. Subasi F. Practical Machine Learning for Data Analysis Using Python. Academic Press. – 2020. – pp. 91-202.
3. Shrouti D., Bayas A., Joshi N., Misal M., Mahajan S., Gite S. Story Generation Using GAN, RNN and LSTM. In: Garg D., Rodrigues J.J.P.C., Gupta S.K., Cheng X., Sarao P., Patel G.S. (eds) Advanced Computing. IACC 2023. Communications in Computer and Information Science. – vol. 2053. – 2024. https://doi.org/10.1007/978-3-031-56700-1_16, pp. 193-204.
4. Li Y., Pan Q., Wang S., Yang T., Cambria, E. A Generative Model for category text generation. Information Sciences. – 2018. – Volume 450. – pp. 301-315.
5. Dychka I., Legeza V., Oleshenko L., Bohutskiy D. Advances in Computer Science for Engineering and Education III, 2020. – pp. 344-346.
6. Tee T.H., Bei Yeap B.Q., Gan K.H., Tan T.P. Learning to Automatically Generating Genre-Specific Song Lyrics: A Comparative Study. In: Villazón-Terrazas B., Ortiz-Rodríguez F., Tiwari S., Sicilia M.A., Martín-Moncunill D. (eds) Knowledge Graphs and Semantic Web. KGSWC 2022. Communications in Computer and Information Science, vol 1686. https://doi.org/10.1007/978-3-031-21422-6_5, 2022. – pp. 62-75.
7. Dhall I., Vashisth S. Micro-Electronics and Telecommunication Engineering. 2020. – pp. 649-657.
8. Goodfellow I.J., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville F., Bengio Y. Generative Adversarial Nets. Département d'informatique et de recherche opérationnelle, Université de Montréal, Montréal, 2014. – p. 9.
9. Kusner M., Hernández-Lobato J.M. GANS for Sequences of Discrete Elements with the Gumbel-softmax Distribution, 2016. – p. 6.
10. Yu L., Zhang W., Wang J., Yu Y. SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient, Shanghai Jiao Tong University, 2016. – pp. 2852-2858.
11. Zuev S.V., Kabalyanc P.S., Polyakov V.M. Detection of stream anomalies by means of the fractal dimension of the graph corresponding to the data processing neural network // Information Systems and Technologies, 2021. – № 5(127). – pp. 31-38.

Колесников Владимир Дмитриевич, аспирант кафедры программного обеспечения вычислительной техники и операционных систем

Кабальянц Петр Степанович, кандидат технических наук, доцент кафедры программного обеспечения вычислительной техники и операционных систем

Kolesnikov Vladimir Dmitrievich, Postgraduate Student Department of Software for Computing Equipment and Operating Systems

Kabalyants Petr Stepanovich, Candidate of Technical Sciences, Associate Professor of the Department of Software for Computing Equipment and Operating Systems