

# Information Technology for Generating Lyrics for Song Extensions Based on Transformers

## **Oleksandr Mediakov**

Lviv Polytechnic National University, Lviv, 79013, Ukraine

E-mail: [oleksandr.mediakov.sa.2019@lpnu.ua](mailto:oleksandr.mediakov.sa.2019@lpnu.ua)

ORCID iD: <https://orcid.org/0000-0002-2580-3155>

## **Victoria Vysotska**

Lviv Polytechnic National University, Lviv, 79013, Ukraine

Osnabrück University, Osnabrück, 49076, Germany

E-mail: [Victoria.A.Vysotska@lpnu.ua](mailto:Victoria.A.Vysotska@lpnu.ua), [victoria.vysotska@uni-osnabrueck.de](mailto:victoria.vysotska@uni-osnabrueck.de)

ORCID iD: <https://orcid.org/0000-0001-6417-3689>

## **Dmytro Uhryn**

Yuriy Fedkovych Chernivtsi National University, Chernivtsi, 58012, Ukraine

E-mail: [d.ugryn@chnu.edu.ua](mailto:d.ugryn@chnu.edu.ua)

ORCID iD: <https://orcid.org/0000-0003-4858-4511>

## **Yuriy Ushenko\***

Yuriy Fedkovych Chernivtsi National University, Chernivtsi, 58012, Ukraine

E-mail: [y.ushenko@chnu.edu.ua](mailto:y.ushenko@chnu.edu.ua)

ORCID iD: <https://orcid.org/0000-0003-1767-1882>

\*Corresponding Author

## **Cennuo Hu**

Department of Computer Science, College of Science, Purdue University, West Lafayette, IN 47907, USA

E-mail: [hu945@purdue.edu](mailto:hu945@purdue.edu)

ORCID iD: <https://orcid.org/0009-0008-9589-9253>

Received: 23 October, 2023; Revised: 12 November, 2023; Accepted: 16 December, 2023; Published: 08 February, 2024

**Abstract:** The article develops technology for generating song lyrics extensions using large language models, in particular the T5 model, to speed up, supplement, and increase the flexibility of the process of writing lyrics to songs with/without taking into account the style of a particular author. To create the data, 10 different artists were selected, and then their lyrics were selected. A total of 626 unique songs were obtained. After splitting each song into several pairs of input-output tapes, 1874 training instances and 465 test instances were obtained. Two language models, NSA and SA, were retrained for the task of generating song lyrics. For both models, t5-base was chosen as the base model. This version of T5 contains 223 million parameters. The analysis of the original data showed that the NSA model has less degraded results, and for the SA model, it is necessary to balance the amount of text for each author. Several text metrics such as BLEU, RougeL, and RougeN were calculated to quantitatively compare the results of the models and generation strategies. The value of the BLEU metric is the most diverse, and its value varies significantly depending on the strategy. At the same time, Rouge metrics have less variability and a smaller range of values. In total, for comparison, we used 8 different decoding methods for text generation supported by the transformers library, including Greedy search, Beam search, Diverse beam search, Multinomial sampling, Beam-search multinomial sampling, Top-k sampling, Top-p sampling, and Contrastive search. All the results of the lyrics comparison show that the best method for generating lyrics is beam search and its variations, including ray sampling. The contrastive search usually outperformed the usual greedy approach. The top-p and top-k methods do not have a clear advantage over each other, and in different situations, they produced different results.

**Index Terms:** Transformers, T5 language model, recurrent networks, text generation, author's style

## 1. Introduction

Today, pre-trained large language models are the driving force behind the development of not only NLP but also deep learning systems in general. Transformer models [1] are capable of solving virtually all tasks

In turn, words, sentences, and texts are the basic and most important way of communication between intellectually developed beings. Of course, speech and texts are used to convey certain emotions, events, etc. One of the main ways language is used to describe emotions is through songs with lyrics.

However, often, due to the need to preserve rhyme and rhyme, the dimensionality of the verse lines, the structure of the song, etc., artists have to use repetition of lines in the lyrics. In addition, the process of writing lyrics can be lengthy.

In this article, we propose to consider the problem of generating song lyrics extensions using large language models, in particular the T5 model [2], to speed up, supplement, and increase the flexibility of the songwriting process.

This work aims to build a system for generating song lyrics with the T5 model with and without taking into account the author's style. In accordance with the goal, the main tasks of the work are as follows:

- creation of training and test data samples, their processing, standardization and preparation,
- fine-tuning of two T5 models for the task of generating song lines,
- designing a system for users to use these models,
- experimental testing of the trained machine learning models,
- analysis and discussion of the data obtained as a result of the experimental testing.

The object of research is the process of artificial text generation by T5 transformer models. The subject of the study is the process of retraining language generation models to create continuations of song lyrics (with and without taking into account the style of a particular author).

Creating such a project will have several effects:

- Scientific, technical, and social, which will be expressed in the creation of a new data set (a special form for a specific task, but open to modification), and the publication of trained models;
- Potentially accelerate and increase the flexibility of the process of creating lyrics for songs, as the system will be able to create several variants of continuation ribbons for existing lines;
- Potentially speed up the process of experimenting with different lyrics to create the final composition.

The use of sufficiently powerful and modern language models, in particular T5, will ensure the quality of the task, provided that a high-quality training data set is created, the rules for model retraining are followed, and the correct text generation strategy is chosen. The pure text-to-text approach to the task, which is supported by the T5 model [2], will allow using one model to generate lyrics in the style of many authors since the author of the song can be encrypted in the input text itself. Potentially, this approach allows us to create one model for two subtasks (generation with and without specifying the author's style), but in order to compare the ability to reproduce the authors' lyrics, we decided to build two separate models that will be trained under the same conditions.

## 2. Literature Review

A text generation strategy or a text decoding method in language models is an algorithm for selecting the next word of a sequence from the results of language models, taking into account the previous words. The existence of different approaches to decoding is due to the diversity of tasks, purposes, and architecture of models and the development of methodologies in NLP [3, 4].

In the scientific world, it is common to experiment with different text generation strategies, selecting and studying the best one for a particular task. In this article, we consider two language transformer models T5 [2, 5], which have been trained for the task of generating song lyrics. These models solve the same task but with different approaches. The aim of this article is to compare the quality of reproduction of generated lyrics according to special textual metrics for the two models using different generation strategies.

The first model, which will be referred to as NSA (from a non-specific author), is trained to generate song continuations without any conditions, and the second model, SA (from a specific author), solves the task of generating continuations when the author's style is specified. For the second model, the author of a particular set of lines and their continuation is added as an additional parameter during training. During generation, certain additional restrictions and parameters are set, including a string length limit of 8 to 128 tokens, a softmax function temperature of 0.98, and a repetition penalty [3] of 0.98. All other settings are specific to each strategy.

It is worth noting that in the following, the concepts of string, tape, and chain will be identified as a sequence of elements, including words. In this case, the concept of a word as an element of a chain can be replaced by a token or a hypothesis, or a candidate.

This replacement is possible because, in this work, the index of a word in the dictionary is considered a token.

When analyzing analogues to the system being developed in this article, it is necessary to take into account both close and distant analogues that solve the same problem, as well as analogues of language models that can be used for the purpose.

The main indirect analogs for the system for comparing generation strategies are other projects that allow you to create continuations of song lyrics. For example:

- MuseNet by OpenAI: MuseNet is a music language model that can generate continuations of musical compositions.
- Jukedeck: Jukedeck is a platform that uses artificial intelligence to automatically generate music. It is capable of creating music extensions based on style, mood, and other parameters.
- Magenta Project by Google: Magenta is an open source for researching creativity, music, and machine learning. It includes various models and tools for generating music, including the generation of lyrics for song sequences.
- Amper Music: Amper Music is a platform that uses artificial intelligence to automatically generate music for videos and other media projects. It can generate music sequences based on specified parameters.

However, a conceptually higher level of analogues that needs to be considered is the type of neural networks and models that can be used for NLP tasks, and in particular for text generation. We can identify the following set of close competitor models that can be used for text generation:

- CNN or convolutional neural networks;
- RNN or recurrent networks;
- Transformers or transformer models.

Convolutional neural networks are the least adapted to the task of text generation (although they are used for text classification tasks, etc.), which is quite obvious given their inability to work with streaming data and the nature of convolution.

Recurrent networks are the first qualitative alternative since this type of neural network allows solving the problem of sequentially generating song lyrics. However, there are several aspects in which recurrent networks lose out to transformers [1]:

- Recurrent networks are not able to store long-term relationships in the text, which is very important for generating coherent lyrics, while transformers are able to do this due to the nature of the attention mechanism;
- The process of training recurrent networks is limited, as they suffer from the problem of gradient fading, and their work cannot be parallelized, which is not a problem for transformers;
- It is not an easy process to add additional parameters to the input text, for example, the desired text steel.

However, transformer models also have their drawbacks, which can affect the system's performance, for example:

- The need for extremely large amounts of data for high-quality work, but this problem can be solved by using transfer learning, i.e., using a pre-trained model and training it for a specific task (in this case, generating song lyrics);
- Some models tend to generate uncreative text, such as the first version of T5 [7];
- The quality of the network depends on the quality of the data and the organization of the training process;
- The need for a large amount of computing resources, has an impact on the planet's ecology [8].

Of course, the impact of most disadvantages can be minimized by appropriate actions and implementation of transformers' pre-training practices. In addition, the use of transfer training and transformers, in particular the T5 model, provides the system with many advantages, namely:

- Contextual generation, i.e., taking into account the context of the input text strings to generate output;
- Ease of implementation of conditional generation, with an indication of the desired style, and the direct possibility of studying the properties of authorship;
- The ability to upgrade to higher quality or newer versions of specific transformer architectures.

Given the above analysis, it is the transformer models using transfer learning that are the most optimal option for achieving this goal.

### 3. Material and Methods

For the correct construction of the system and its specification, it is necessary to define the main definitions, terms, and objects of the subject area. However, the scope of artificial text generation is quite wide, so it is necessary to define the boundaries of the subject area. Fig. 1 shows the precedent diagram of the proposed system. The role of this diagram is to show what functional requirements the system should have in order to be able to identify the boundaries of the subject area.

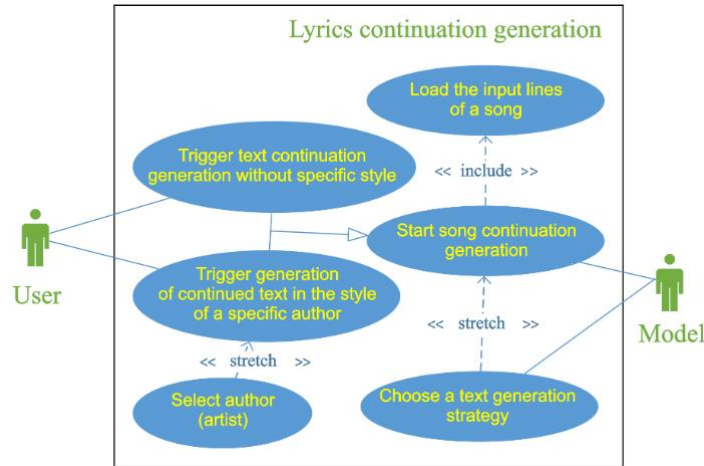


Fig. 1. Diagram of the precedents of the system for generating song lyrics continuation

The diagram shows that conceptually, the highest precedent of the system is processing a user's input request to generate a continuation of the lyrics. Given the described precedents, as well as the peculiarities of developing a system based on large language models, three important aspects of the subject area can be identified:

- Training dataset creation is the process of processing and preparing textual data;
- Fine-tuning a language model is the process of training a pre-trained model to perform a specific task;
- Song lyrics generation - the process of using the model and decoding its results.

Each of these aspects contains specific timelines and processes and options that need to be used to achieve the goal.

Creating a training dataset is the most important step for systems using artificial intelligence, as data quality is a prerequisite for building a model that meets the requirements.

From this part of the subject area, it is necessary to distinguish what it means to generate a continuation of the song lyrics. This question arises because this task can be defined in several ways. For example, the classic task of language models is to recover missing (masked) words or n-grams in a text, the so-called language modelling. The same principle can be applied to the process of generating song sequences, i.e., masking line endings or random n-grams from the lyrics, and training the model to restore them. This approach requires potential users to provide not only the text but also the places where they need to complete the text. Instead, this article proposes a different approach - line-by-line completion of the lyrics. Instead of continuing each line individually, the model will try to play several subsequent lines. To do this, a special line break token will be added to the text, and the model will try to generate it at the point where the line ends.

The second aspect of the domain is the process of fine-tuning the language model. The main goal of the process is to improve the model results and provide more accurate and appropriate lyrics generation for song continuation. This aspect includes three steps: creating the specified data (described earlier), selecting the architecture, and reconfiguring the model.

For the context of this article, it is important to note the second step - the choice of architecture. This process is complicated, as there are many different models with different sizes, layers, and a number of parameters. When choosing an architecture, one should take into account the peculiarities of the task, available computing capabilities, available memory resources, and the impact of the model training process on the environment. The last aspect to consider is the actual process of generating text with a language model.

To describe this process, we need to introduce several definitions.

- Tokenization is the splitting of tapes into separate components (tokens), which can be words, symbols, or word fragments. At the same time, encoding during tokenization is the process of mapping tokens to their numerical representation.

- Decoding tokens is the process of mapping indices to the corresponding parts of speech (words, symbols, etc.).
- The generation strategy or decoding strategy [3] is an algorithm for selecting the next word, depending on the previous ones. The choice of strategy affects the direct use of the model during generation and has a significant impact on the final result, compatibility, and quality of the generated text.

The entire process of generating lyrics for songs within the context of the formed context can be represented as a sequence of operations, formalized in the activity diagram shown in Fig. 2. After introducing the main terms and subprocesses, it is possible to form a description of the main process of the system - processing a request for generating a song lyric. Fig. 3 shows the corresponding activity diagram. The objects used in Fig. 3 are abstractions that do not require specification.

The description of the selected methods and tools for system development can be divided into two parts: the first is the tools for solving the generation problem, and the second is software development technologies.

The chosen architecture of the transformer model is T5 [2]. The peculiarity of this model is its completeness (encoder-decoder model), flexibility, and the use of a text-to-text approach. This approach allows unifying one model for several tasks, such as generating lyrics in the style of different authors, by adding a so-called task-specific prefix.

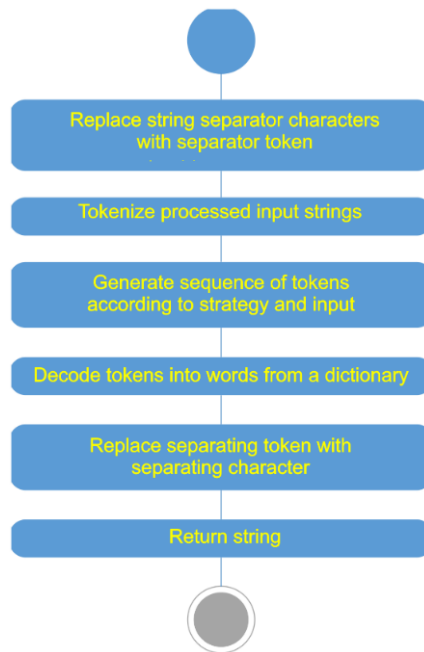


Fig. 2. Activity diagram of the process of generating lyrics for a song continuation

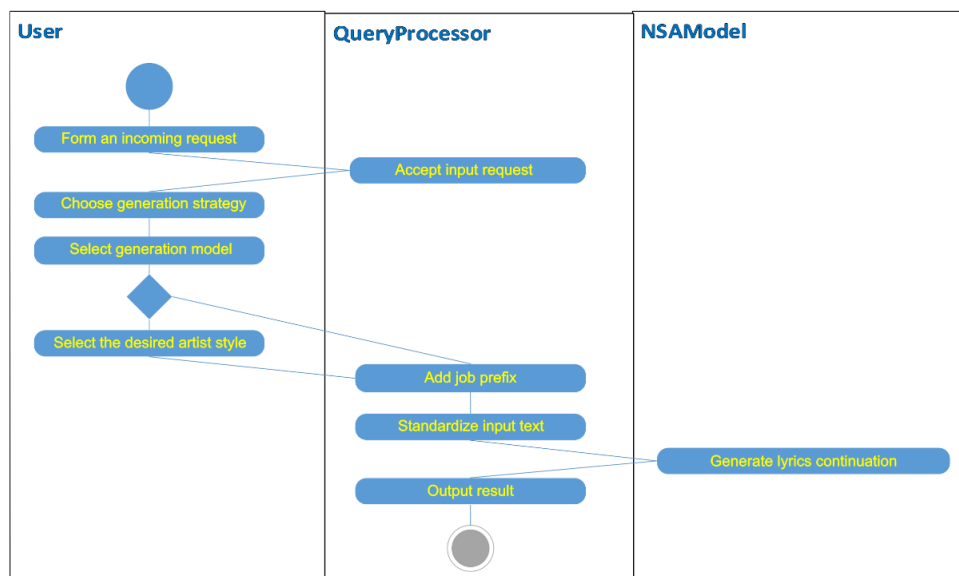


Fig. 3. Activity diagram of the main process - processing a request to generate lyrics for a song sequence

Other advantages of the model:

- Contextual generation, i.e. taking into account the context of the input text to generate the output;
- Targeted generation, i.e. the use of special tokens and masks;
- A range of different sizes, allows you to choose a model that fits your computational constraints and needs.

The corresponding tokenizer and detokenizer for this model are based on the sentencepiece [9], which implements the subword units tokenization model. The use of such a tokenizer has a number of advantages, for example, it is language-independent, lightweight, and very fast [9]. According to the authors of this tokenizer, it is a better alternative for generation tasks than subword-nmt [10] or WordPiece [11].

For the generation process, the choice of decoding strategy is important. However, instead of giving preference to a specific strategy, the system will support a variety of strategies. In particular, the following 8 strategies:

1. Greedy search;
2. Beam search;
3. Diverse beam search;
4. Multinomial sampling (selection);
5. Beam-search multinomial sampling (selection);
6. Top- $k$  sampling;
7. Top- $p$  sampling;
8. Contrastive (comparative) search.

The Python language and a set of necessary libraries were chosen for the direct development, as well as model training, data preparation, and other steps of the system implementation process.

The most important library is transformers [12], which allows you to load, train, and save a T5 model and the corresponding tokenizer. The model itself will be an object from TensorFlow [13].

#### 4. Results. Basic Material

To prepare the data, we first need to obtain it. To do this, we used the database of the Genius service [14] as of the summer of 2021 in the form of a JSON file.

To create the data, 10 different artists were selected, and then their lyrics were selected. A total of 626 unique songs were obtained.

Each song lyric consists of strings of words and reference information (parts of the song structure or which of the collaborators performs a particular lyric). The first step in processing the lyrics is to remove the reference information and break the lyrics into strings.

After preparing the text data, you need to divide the tapes into beginning and end, i.e., into input and output data. Before processing the text, it is necessary to determine how many input-output pairs can be formed from each song. For this purpose, we performed exploratory statistical analysis for the number of ribbons in the songs, as well as the number of words and unique words in the songs. Fig. 4 shows histograms, boxplots, and statistics for each of the characteristics.

Using the data from Fig. 4, we developed a process for splitting tapes into input and output. This process must take into account several requirements, in particular: the input-output pair must be consecutive song lyrics, the length of the input and output tapes must vary from instance to instance, and the number of input and output tapes must be at least a specified minimum.

A special <sep> token is used to combine the tapes of the input and output groups. This token is also added to the tokenizer. As a result of splitting each song into several pairs of input-output feeds, 1874 training instances and 465 test instances were obtained. To illustrate an example of the process of converting a song lyric into one dataset instance, the corresponding image is shown in Fig. 5.

Two language models, NSA and SA, were retrained for the task of generating song lyrics. For both models, t5-base was chosen as the base model. This version of T5 contains 223 million parameters. As previously described, two models were developed in this article. The first one solves the problem of generating song lyrics, and the second one is used to generate lyrics in the style of a particular author. There are 10 authors in total, which corresponds to 10 artists whose songs are selected for training.



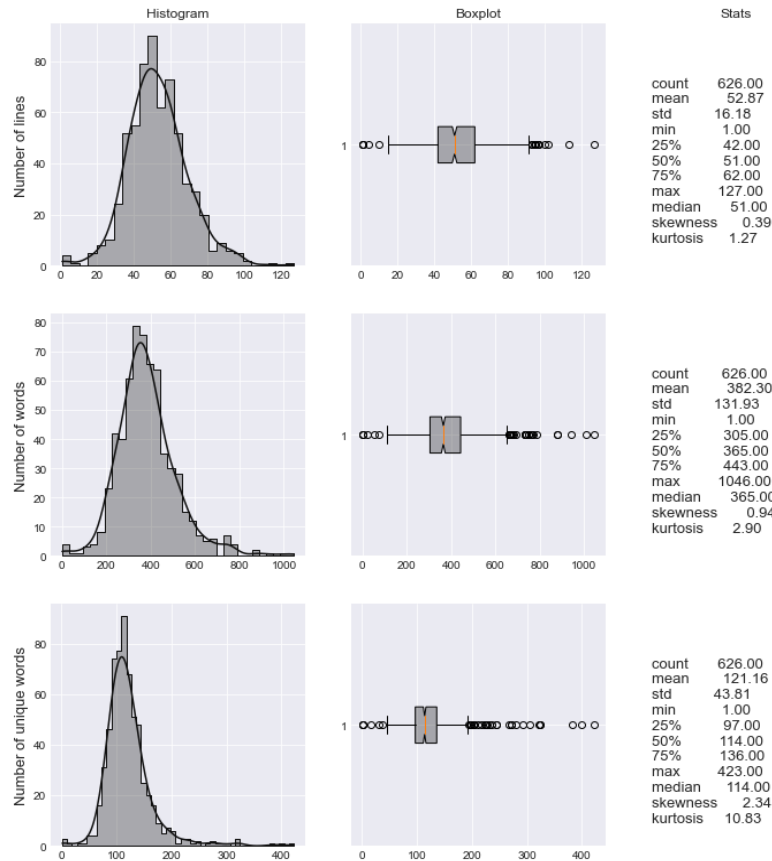


Fig. 4. Statistical analysis of the number of lines, words and unique words in the selected songs

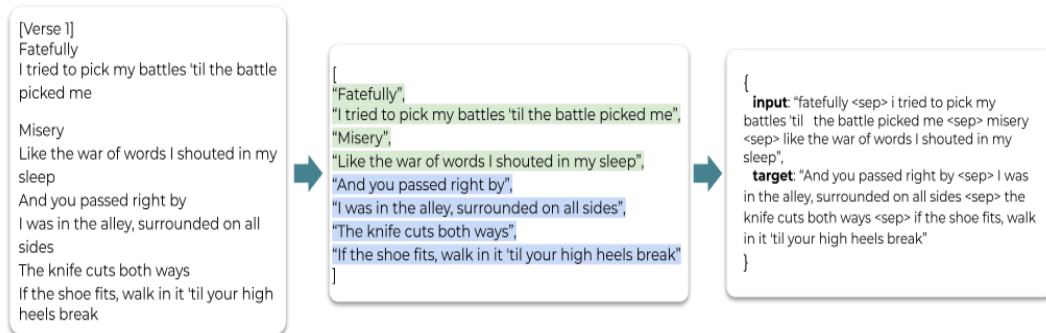


Fig. 5. An example of processing a fragment of song lyrics (clean, split, and ribbons, forming an input-output pair)

For the first model, which will be called NSA (from a non-specific author), the task-specific prefix looks like "continue lyrics:". Accordingly, all input lyrics are prefixed with this prefix before the model is trained.

Accordingly, for the second model, called SA, this prefix depends on the author of the text, but the general form is "continue lyrics as [a certain author]".

Hyperparameter settings for models are the same:

- Number of epochs: 3;
- Data batch size: 2 records;
- Optimizer: Adam with a learning rate of  $5 \cdot 10^{-5}$ .

The training metrics are the value of the loss function, the recall, and the top-3 accuracy. The latter metric determines the accuracy of predicting the required token, taking into account the three most likely outcomes. The training results for each model are presented in the form of graphs of learning curves, for the NSA model - Fig. 6, and for the SA model - Fig. 7.

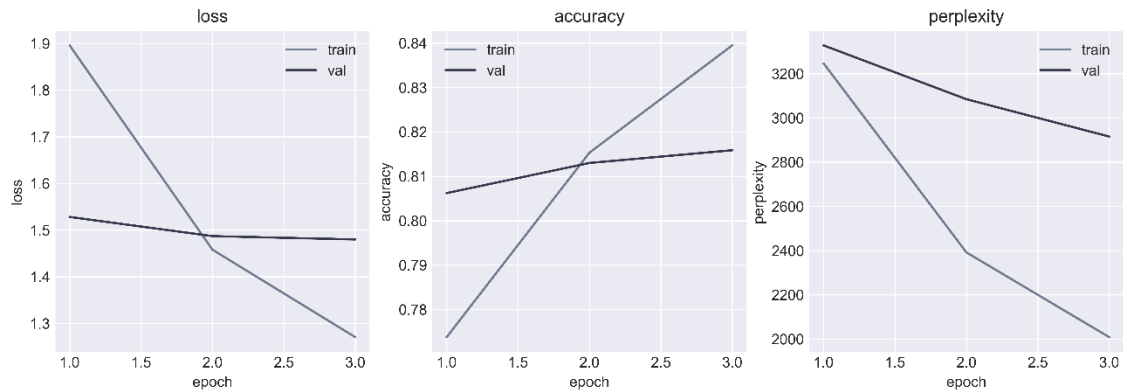


Fig. 6. Learning curves of the NSA model

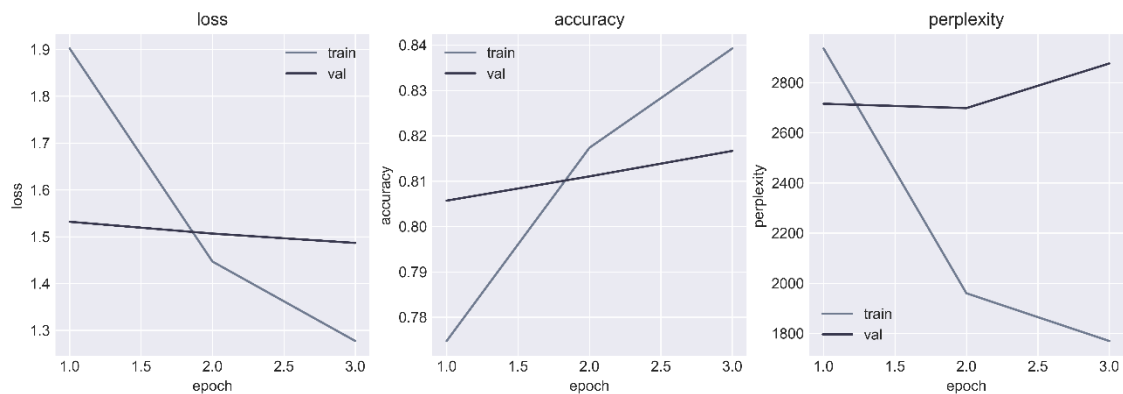


Fig. 7. Learning curves of the SA model

The analysis of the learning curves indicates that the NSA model will have less degraded results, while the SA model needs to balance the amount of text for each author.

After developing and training the models, it is possible to use the models for text generation. Two options are considered as benchmark examples: the first is to consider the directly generated text for a specific input query, and the second is to quantify the quality of the model for a set of input queries.

The input tape for example, by T. Swift and A. Desner:

*"Past me*

*I wanna tell you not to get lost in these petty things your nemeses will defeat themselves before you get the chance to swing and he's passing by rare as the glimmer of a comet in the sky and he feels like home if the shoe fits, walk in it everywhere you go and I fell from the pedestal right down the rabbit hole"* (fragment from "long story short" [15]).

In total, for comparison, the article uses 8 different decoding techniques to generate text supported by the transformers library.

1. Greedy search;
2. Beam search;
3. Diverse beam search [6];
4. Multinomial or polynomial sampling (selection);
5. Beam-search multinomial sampling (selection);
6. Top-k sampling [16];
7. Top-p sampling [17];
8. Contrastive or comparative search [18-21].



Methods (research strategies) 1, 2, 3, and 8 are deterministic, and 4-7 are stochastic.

Although it is possible to choose a specific decoding strategy, the benchmark example considers all 8 of them. For both models, the same input tape, the same repetition penalties, the so-called softmax function temperature, and the same parameter settings of the generation strategies are used.

The results of the NSA model are shown in Fig. 8-9, and the results of the SA model are shown in Fig. 10-11.

For the second benchmark example, a set of input-output pairs of tapes is selected, and several textual metrics are calculated to quantitatively compare the results of the models and generation strategies. These metrics are BLEU, RougeL, and RougeN. Fig. 12 shows the metric values for the ultra-large sample from the test data for the NSA model since its results are better than those of the SA model (Fig. 13). All the results from Fig. 8 and 9 do not contain text repetition, but some of the texts are grammatically incorrect, and some have illogical word combinations. Beam search, top-p, and diversified beam search contain abnormally long strings.

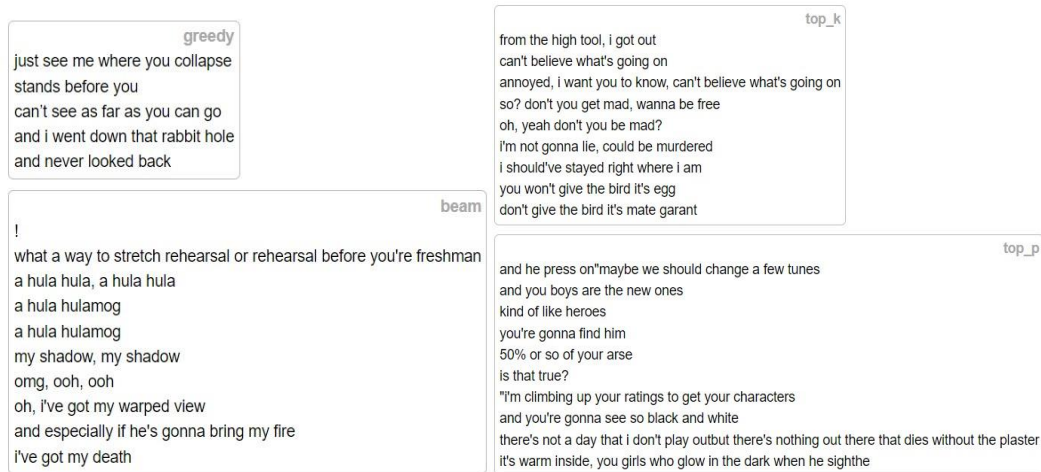


Fig. 8. Results generated by the greedy search, beam search, top-p, and top-k strategies

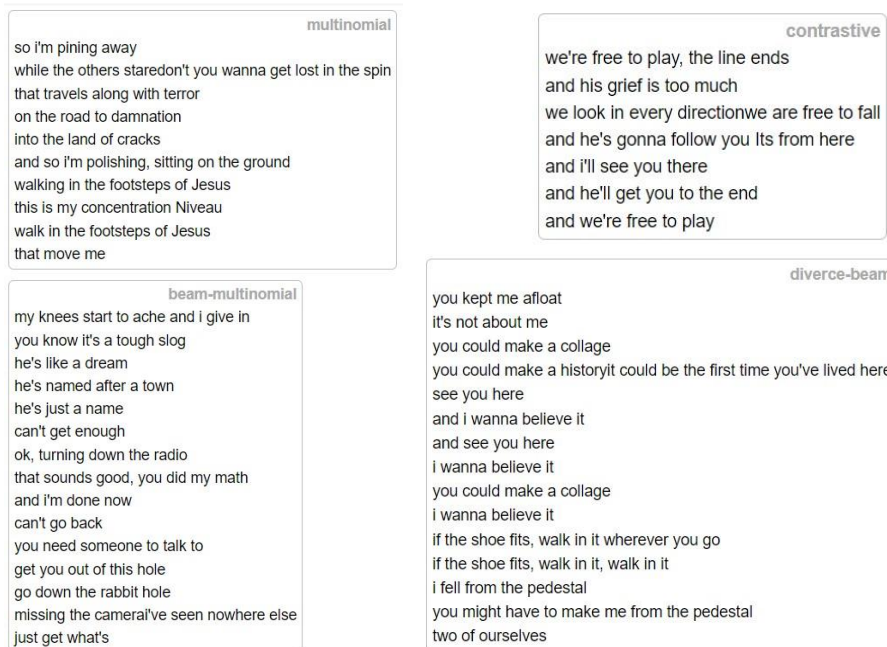


Fig. 9. Results generated by sampling, beam sampling, diverse-beam, and contrast searches

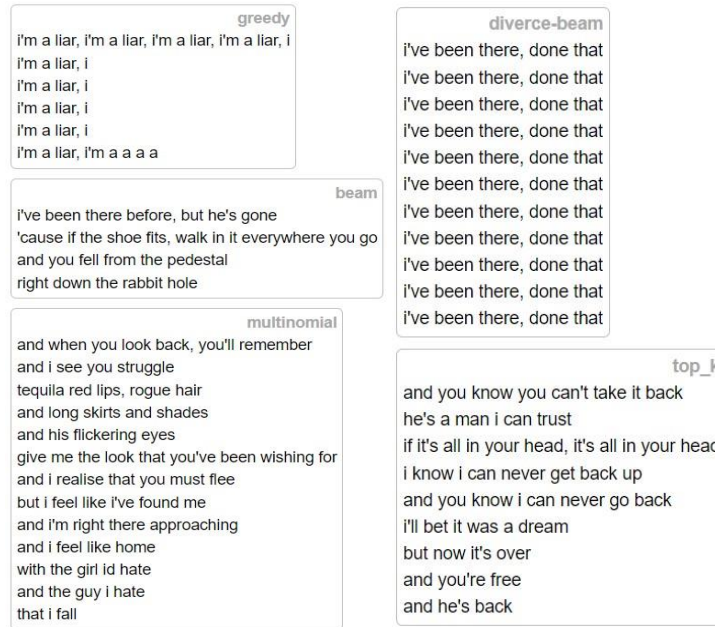


Fig. 10. Results of generating 5 strategies from the SA model

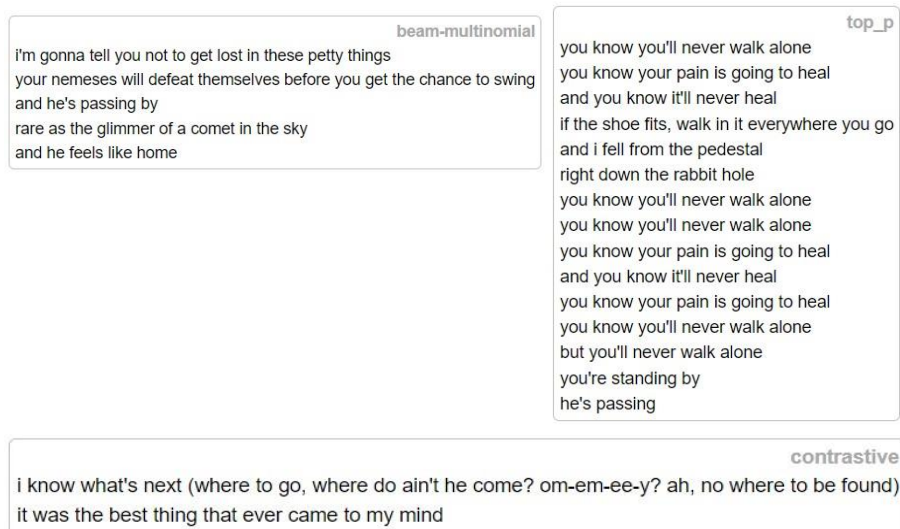


Fig. 11. Results of generating 3 strategies from the SA model

The results from the SA model show visible signs of text degradation, repetition, and inconsistency. These problems are probably signs of a revision of the learning process.

Fig. 12 shows how changing the strategy changes the value of the metric. In this case, the ray sampling strategy has the best results. In the context of the developed system, two characteristics are the most important for analysis. The first is the estimation of the number of emissions and impact on the planet's ecology through the use and training of models on hardware accelerators. To estimate this characteristic, we can use the idea from [8].

Thus, at the time of the work, taking into account the experiments and the use of models, training two T5 models on the T4 accelerator in the Google Colab environment, an average of almost 0.55 kg of carbon dioxide (carbon dioxide) was emitted.

The second characteristic is directly related to the first one: the time taken to generate and decode the text using different strategies. To analyze this time, we recorded the time of text computation and decoding during the generation of examples. Of course, this time depends on the size of the received tape, other system processes, etc. Therefore, before visualization, the obtained time is normalized by the maximum value.

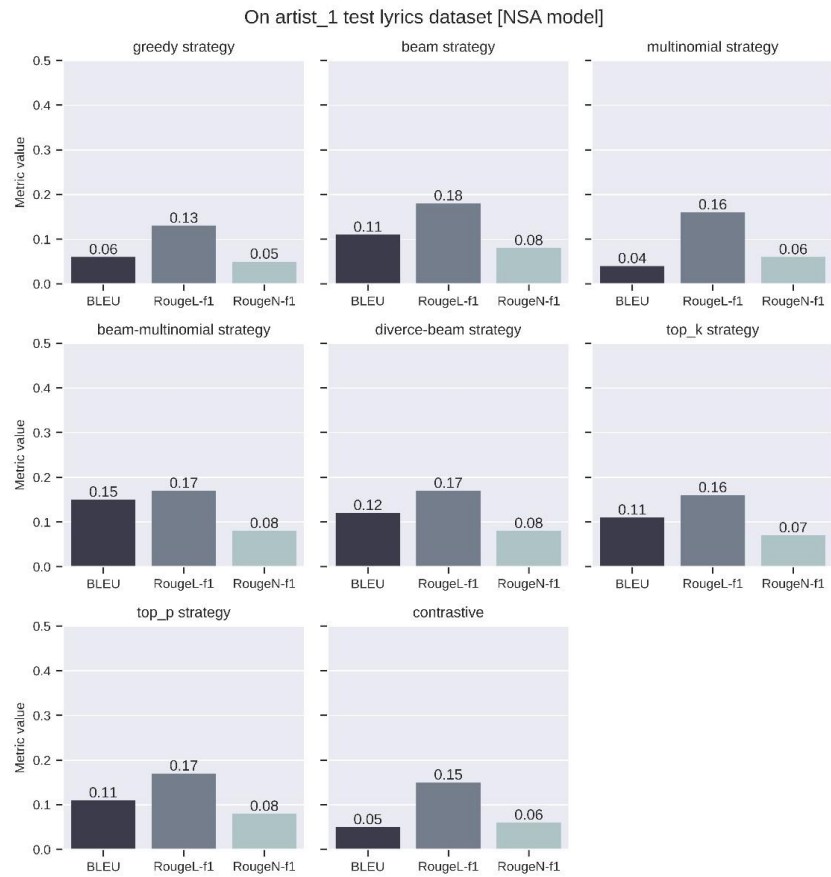


Fig. 12. Values of text similarity metrics from the NSA model

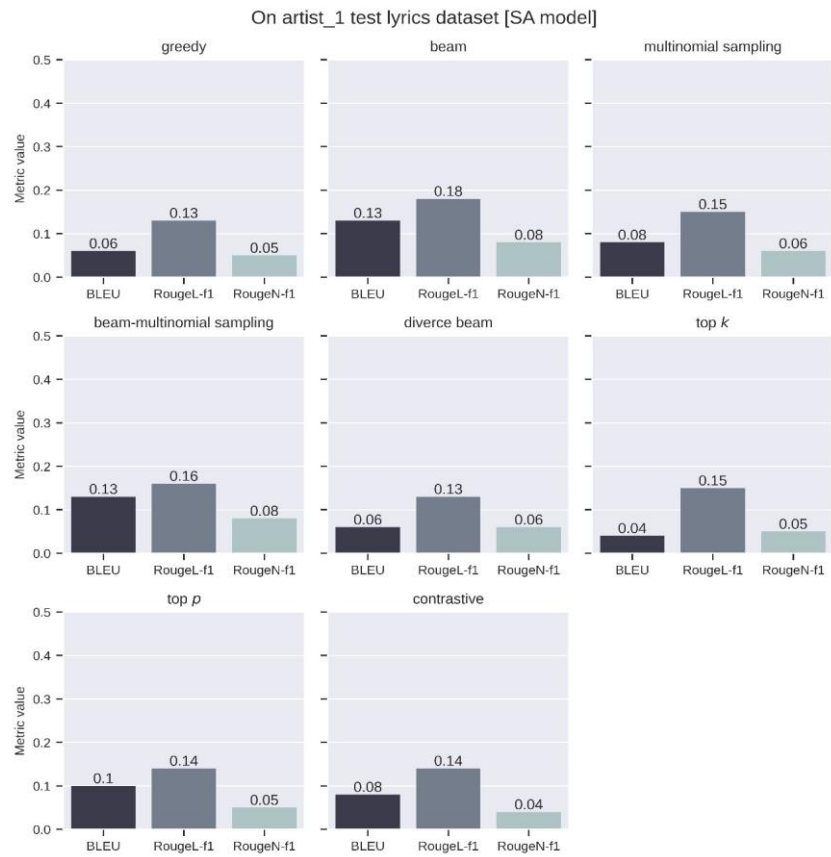


Fig. 13. Values of text similarity metrics from the SA model

Fig. 14 shows the conditional time required to generate results by the NSA model, and Fig. 15 shows the SA model with small amounts of input data.

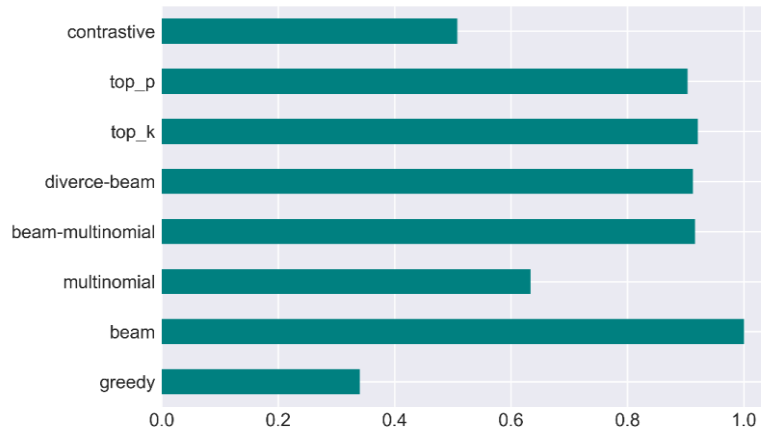


Fig. 14. Normalized generation execution time by the NSA model

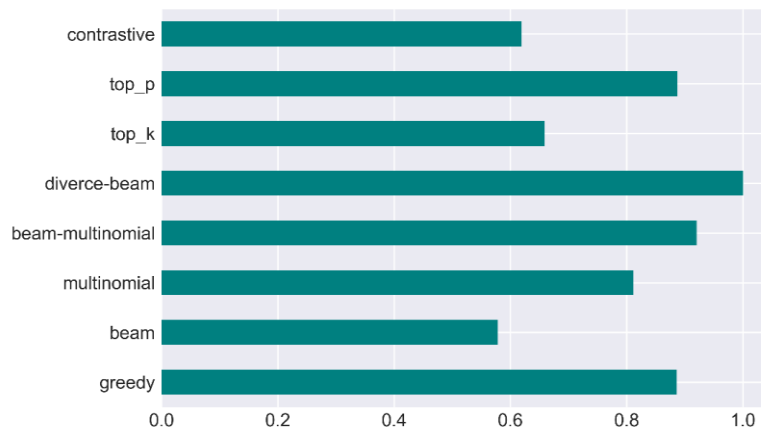


Fig. 15. Normalized generation execution time by the SA model

In both cases, beam search variations are the longest in terms of execution time, and contrast search is one of the shortest.

## 5. Discussion and Conclusions

When formulating the context of this work, its main goal is to create a system for generating continuation of song lyrics from and bases for taking into account the author's style. As a result of this work, this goal has been achieved because:

- A systematic analysis of the subject area in which the system operates is carried out. The basic concepts, processes, and necessary terms that allow to create and implement the described system are identified;
- The tasks of creating a textual dataset of input-output pairs of song feeds were completed;
- The fine-tuning of two transformer models of the T5 architecture was carried out to solve the problem of artificial generation of the song lyrics with and without taking into account the author's style;
- The impact of the training process on the environment is analyzed, and the generation time is estimated, depending on the decoding strategy;
- Examples of generating song sequences are given, and the resulting text is analyzed;
- The quality of text generation by the created models is quantitatively evaluated using the metrics of tape comparison.

It is logical that the created system has ways to develop and improve. The main prospects for improving the system are those related to training data, transformer models, and the development process itself. Therefore, we can identify the following areas for future research:

- Using newer or more powerful models than the classic T5;
- Expanding the amount and variety of training data. Replacing or combining several types of song lyrics generation tasks;
- Expanding the group of languages supported by the system;
- Including other structural elements of a song (rhythm, chords, verse line dimension) or musical accompaniment of a corresponding song as input for lyrics generation;
- Improving the model training process, for example, by applying a contrastive training process or refinement, i.e. involving real authors to assess the quality and improve the model results.

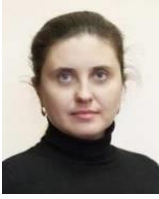
## References

- [1] Ashish Vaswani and others. Attention Is All You Need. URL: <https://arxiv.org/abs/1706.03762>
- [2] Colin Raffel and others. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. URL: <https://arxiv.org/abs/1910.10683>
- [3] Text generation strategies. URL: [https://huggingface.co/docs/transformers/v4.29.0/en/generation\\_strategies](https://huggingface.co/docs/transformers/v4.29.0/en/generation_strategies)
- [4] How to generate text: using different decoding methods for language generation with Transformers. URL: <https://huggingface.co/blog/how-to-generate>
- [5] T5. Overview. URL: [https://huggingface.co/docs/transformers/model\\_doc/t5](https://huggingface.co/docs/transformers/model_doc/t5)
- [6] Ashwin K Vijayakumar and others. Diverse Beam Search: Decoding Diverse Solutions From Neural Sequence Models. URL: <https://arxiv.org/abs/1610.02424>
- [7] T5v1.1. URL: [https://huggingface.co/docs/transformers/model\\_doc/t5v1.1](https://huggingface.co/docs/transformers/model_doc/t5v1.1)
- [8] Lacoste A., Luccioni A., Schmidt V., Dandres T. Quantifying the Carbon Emissions of Machine Learning. URL: <https://arxiv.org/abs/1910.09700>
- [9] Google's SentencePiece. URL: <https://github.com/google/sentencepiece>
- [10] Subword Neural Machine Translation. URL: <https://github.com/rsennrich/subword-nmt>
- [11] Yonghui Wu and others. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. URL: <https://arxiv.org/pdf/1609.08144.pdf>
- [12] Transformers. State-of-the-art Machine Learning for PyTorch, TensorFlow, and JAX. URL: <https://huggingface.co/docs/transformers/index>
- [13] Martín Abadi and others. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. URL: <https://www.tensorflow.org/>
- [14] Song Lyrics Dataset. URL: <https://www.kaggle.com/datasets/deepshah16/songlyrics-dataset>
- [15] Swift T., Desner A. "long story short". Taylor Swift Music, 2020.
- [16] Fan A., Lewis M., Dauphin Y. Hierarchical Neural Story Generation. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Melbourne, Australia. Stroudsburg, PA, USA, 2018. URL: <https://doi.org/10.18653/v1/p18-1082>
- [17] Chiang T.-R., Chen Y.-N. Relating Neural Text Degeneration to Exposure Bias. Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP, Punta Cana, Dominican Republic. Stroudsburg, PA, USA, 2021. URL: <https://doi.org/10.18653/v1/2021.blackboxnlp-1.16>
- [18] Yixuan Su and others. A Contrastive Framework for Neural Text Generation. URL: <https://arxiv.org/abs/2202.06417>
- [19] Ihor Tereikovskiyi, Zhengbing Hu, Denys Chernyshev, Liudmyla Tereikovska, Oleksandr Korystin, Oleh Tereikovskiyi, "The Method of Semantic Image Segmentation Using Neural Networks", International Journal of Image, Graphics and Signal Processing, Vol.14, No.6, pp. 1-14, 2022.
- [20] Chandra Shekhar Tiwari, Vijay Kumar Jha, "Enhancing Security of Medical Image Data in the Cloud Using Machine Learning Technique", International Journal of Image, Graphics and Signal Processing, Vol.14, No.4, pp. 13-31, 2022.
- [21] Ramesh M. Kagalkar, "Methodology for Translation of Video Content Activates into Text Description: Three Object Activities Action", International Journal of Image, Graphics and Signal Processing, Vol.14, No.4, pp. 58-69, 2022.

## Authors' Profiles



**Oleksandr Mediakov** is an accomplished senior student pursuing his undergraduate degree in the Department of Information Systems and Network at Lviv Polytechnique National University. He is a budding researcher passionate about Artificial Intelligence, particularly Machine and Deep Learning.



**Victoria Vysotska** is Postdoctoral Researcher of Osnabrück University and an Associate Professor at Information Systems and Networks Department of Lviv Polytechnic National University. She received the PhD degree in Information Technologies from Lviv Polytechnic National University in 2014. She has currently published more than 450 publications. Her main research interests are focused on NLP, computer linguistics, data science, system analysis, information technologies, machine learning.



**Dmytro Uhryn** graduated from Yuriy Fedkovych Chernivtsi National University, Chernivtsi. Currently, he is Doctor of Technical Sciences, Associate Professor Yuriy Fedkovych Chernivtsi National University. He has currently published more than 140 publications. His research interests are data mining, information technologies for decision support, swarm intelligence systems, industry-specific geographic information systems.



**Yuriy Ushenko:** Prof., Computer Science Department, Chernivtsi National University, Chernivtsi, Ukraine. Research Interests: Data Mining and Analysis, Computer Vision and Pattern Recognition, Optics & Photonics, Biophysics.



**Cennuo Hu** was born in 2005. Now she is a student in the Department of Computer Science of the College of Science at Purdue University, West Lafayette, IN 47907, USA. Her main research interests are software engineering and computer security. She is an IEEE student member and has three invention patents.

**How to cite this paper:** Oleksandr Mediakov , Victoria Vysotska, Dmytro Uhryn, Yuriy Ushenko, Cennuo Hu, "Information Technology for Generating Lyrics for Song Extensions Based on Transformers", International Journal of Modern Education and Computer Science(IJMECS), Vol.16, No.1, pp. 23-36, 2024. DOI:10.5815/ijmecs.2024.01.03