Computer Communications 168 (2021) 33–53

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Singability-enhanced lyric generator with music style transfer

Jia-Wei Chang [a], Jason C. Hung [a,*], Kuan-Cheng Lin [b]

[a] National Taichung University of Science and Technology, Taichung City, Taiwan
[b] National Chung Hsing University, Taichung City, Taiwan

ARTICLE INFO

Keywords:
Music style transfer
Lyric generator
GPT-2
Natural language processing

ABSTRACT

The lyrics generator should consider the context and the singability of the songs because every song expresses a story through the context of lyrics, and the lyrics should sound with the music well. Therefore, this study proposes a framework to generate the singable lyrics, and the context of lyrics should fit the given musical style. For the context, this study adopts the GPT-2 model which is powerful for text generation. The conditional GPT-2 model can be used to generate lyrics according to the given style. For suitable for singing, this study adjusts the structure and rhyme of lyrics through the use of a syntactic parser and a rhyme modification module. With automatic and human evaluations, the experimental results show that the proposed method can generate lyrics with high structural consistency, rhyme consistency, and originality according to the given music style.

1. Introduction

Style transfer is an important research topic in deep learning and has yielded significant results in many subfields of artificial intelligence, such as computer vision, image processing, and natural language processing. Specifically, style transfer on text is an important part of natural language generation and paraphrasing, i.e., expressing the same idea or presenting the information in another way [1]. Thus, it facilitates many natural language processing (NLP) applications, such as natural language generation (NLG) [2], machine translation [3], and other natural language interfaces. Current textual style transfer methods require a large parallel corpus for style transfer, and collecting parallel annotations is time-consuming and costly. Most unsupervised text style transfer methods can be divided into two parts. First, separate content from the original style, then blend the content with the applied style. Second, specific style attribute words are removed from the input, and a neutral sequence containing only content words is then fed back to the style-dependent generation model. However, each method has drawbacks: the former tends to change only the style and fails to preserve the content, as it is difficult to obtain style-independent content vectors without parallel data; whereas the latter usually ignores sentiment, which can distort the meaning of the original text. The following problems were identified regarding current research.

- Previous approaches used a more general model for textual style transfer without using novel model concepts.
- Previous studies of textual style conversion used general textual content as input, e.g. newspapers and magazines, with few studies considering lyric texts.
- Style and content are used as latent variables to create a generative framework with style and key sentence conditions included in the model design.
- Consider the structure of the original lyrics to ensure stylized lyrics can have the same structure as the original.
- The method is flexible and capable of different types of style transformations, as evidenced by its success with natural language generation.

This paper proposes a pretrained transformer-2 (GPT-2) based framework for text style transfer. Stylized output is generated from the original lyrics and target style, so we only need to train a decoder to form the different lyric style outputs. This study also considers the original lyrical structure and rhythm. Two post-processing modules were created: a dependency parser to analyze dependencies in each sentence and a rhyme modification to modify sentence endings. For example, suppose the original lyrics have a "pop" style and the goal is to generate lyrics with a "rock" style. Then the rock lyrics dataset is trained for model migration and the lyric text is modified using a post-processing module to ensure that every line and word in the lyrics matches the audio. This study proposes an unsupervised technique based on learning different style characteristics and automatically transferring the lyric text style. The main contributions of this paper can be summarized as follows.

- This study composes a lyrics database to facilitate lyrics style transfer research.

* Corresponding author.
E-mail address: jhung@gm.nutc.edu.tw (J.C. Hung).

https://doi.org/10.1016/j.comcom.2021.01.002
Received 29 August 2020; Received in revised form 23 December 2020; Accepted 9 January 2021
Available online 14 January 2021

- We constructed a lyrics style transfer generation model constructed based on GPT-2 to produce original, thematic, and matched lyrics.
- In contrast with previous studies that fail to fully transfer textual style, the proposed style transfer approach not only shifts at the vocabulary, phrase, or syntax level; but also transforms the musical style. The transformations are considered as a whole.

The remainder of the paper is organized as follows. Section 2 discusses transfer learning, Transformer model, GPT-2, StanfordNLP and provides a literature review of text style transfer. Section 3 describes the lyrics style transfer generation modules, system architecture, model construct, transfer algorithm, etc. Section 4 describe the datasets and evaluation methodology used in experiments and discusses experimental design and results. Section 5 summarizes and concludes the paper, and discusses some directions for future research.

2. Related works

2.1. Transfer learning

Traditional machine learning and deep learning algorithms were generally designed to work in isolation and trained to solve specific tasks. Hence, models must be rebuilt from scratch if the feature space distribution changes. However, most models that solve complex problems require large amounts of data, and acquiring large amounts of annotated data for supervised models is difficult given the time and effort required for annotation, particularly in the context of deep learning. Thus, transfer learning is an important tool in machine learning to address the fundamental problem of insufficient training data. Transfer learning overcomes isolated learning patterns using knowledge learned about a task to solve a related task, i.e., it reuses a model developed for a different task as the starting point for a new task. This allows knowledge to be transferred from existing data to aid future learning. The major advantage is reduced feature extraction and network training time, while avoiding overfitting due to restricted training data. However, one disadvantages include are that training accuracy and generalizability are not guaranteed. Fig. 2-1 shows the transfer learning process.

Deep transfer learning will be widely applied to solve many challenging problems as deep neural networks grow and develop. Golovanov et al. [4] showed that these architectures can be applied and adapted for natural language generation, comparing a number of architectural and training schemes.

2.2. Generative pretrained transformer

The GPT-2 architecture is very similar to the Transformer model decoder structure. However, GPT-2 is a very large Transformer based language model that requires training on a large dataset. The Google Brain team introduced the Transformer [5], which incorporates an encoder–decoder architecture to create a sequence to sequence (Seq2Seq) model without using convolutional (CNN) or recurrent (RNN) neural networks. Traditional CNNs and RNNs were discarded in Transformer, with the entire network composed entirely of attention mechanisms. The encoder comprised 6 coded blocks and the decoder comprised 6 decoded blocks. As with all generative models, encoder output is used as decoder input. Each encoder layer included 2 sublayers containing multi-head self-attention and fully connected feed forward networks. Self-attention can help the current node to not only focus on the current word, but also to obtain contextual semantics. Each decoder layer had 3 sublayers containing multi-head attention, masked multi-head self-attention, and fully connected feed forward networks. This helped the current node obtain highlights that need attention. All sublayers were connected by residuals with dropout to avoid overfitting. Fig. 2-2 shows the Transformer encoder and decoder architecture. Transformer can handle time series and parallel operations, which greatly improved model training probability and the various attention heads can learn different tasks.

Table 2-1
The four GPT-2 model sizes.

Model name	Parameters	Model dimensionality	Layers	Published
GPT-2 Small	117M	768	12	Yes
GPT-2 Medium	345M	1024	24	Yes
GPT-2 Large	762M	1280	36	No
GPT-2 Extra Large	1542M	1600	48	No

GPT-2 incorporates Transformer decoding blocks and outputs one token at a time, similar to traditional language models. The model uses previously outputted tokens as part of the next round of inputs, also known as auto-regression. GPT-2 is trained on a standard task: given a sequence of previous words, predict the next word. Normal self-attention blocks allow the right context to be referenced, whereas GPT-2 uses masked self-attention, where the decoder is only allowed to collect information from prior words in the sentence (plus the word itself). Several recent studies have used these models to extend GPT-2 to specific areas, e.g. Budzianowski et al. [6] proposed task-oriented dialogue bots. Fig. 2-3 shows GPT-2 architecture.

OpenAI has released four GPT-2 model sizes: small, medium, large, and extra-large with 117M, 345M, 762M, and 1542M parameters, respectively, as shown in Table 2-1. More parameters cause increased complexity, larger model size, and higher performance. Training divides into two parts: unsupervised pre-training language model and supervised task fine-tuning. The language model uses multilayered Transformer decoders with multi-head self-attention at input, followed by a feedforward neural network containing location information, and then output a probability distribution for each word. The latter fine-tunes model parameters trained in the previous stage.

GPT-2 pre-training uses the above approach to predict the next word, and hence is better suited for text generation, which commonly generates the next word based on current information. GPT-2 has 1.5B parameters, 10-fold that for the original GPT. The pre-training dataset contains 8 million web pages collected by crawling qualified outbound links from Reddit. Thus, the most important GPT-2 feature is that it uses enough training text to ensure highly significant text generation and hence is a very powerful general purpose language model.

GPT-2 is a Transformer based text generation model, using unsupervised learning to train the language model. The current paper proposes an end-to-end lyric style transfer generation model that can control the output text style using a specific style reference example. The framework applies GPT-2 with several conditions, known as a conditional GPT-2, with architecture as shown in Fig. 2-4.

The generation language model used in this study mainly follows the OpenAI GPT-2 model with some modifications. The model trains a transformer with a 12-layer decoder that has masked self-attention heads (768 dimensions and 12 attention heads) that only focus on information to the left. GPT-2 is essentially a decoder-only transformer, it modifies the transformer's multi-head self-attention mechanism for handling longer sequences, such as song lyrics. The proposed GPT-2 language model divides into three parts: embedding layer, transformer layer, and output layer; where each layer is described in detail below. Implementation details are available online at https://github.com/adigoryl/Styled-Lyrics-Generator-GPT2.

GPT-2 can only generate a maximum of 1024 tokens per request (Transformer only processes 512 tokens), and each token is embedded in the decoder block along its own path. Similar to other NLP models, word embedding is represented by an embedding matrix and the model can be queried in a matrix. We divide the embedding layer into two parts for introduction. Token embeddings (wte) are shown in Fig. 2-5, left. Each line is a word embedding, i.e., a list of numbers representing a word and capturing its partial meaning. The model vocabulary in this study contained 50,269 items and we used the GPT-2 model with

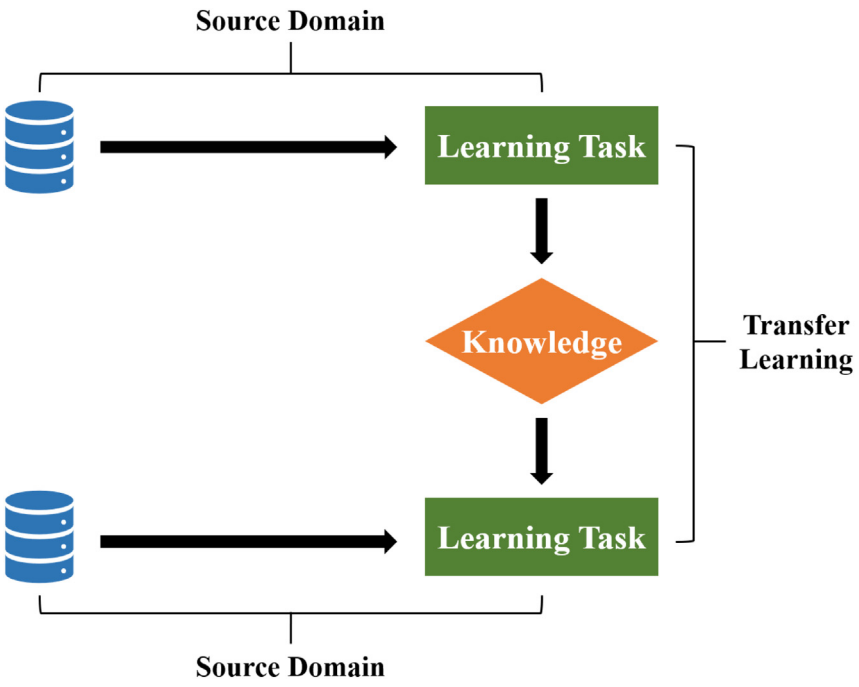


Fig. 2-1. The learning process of transfer learning.

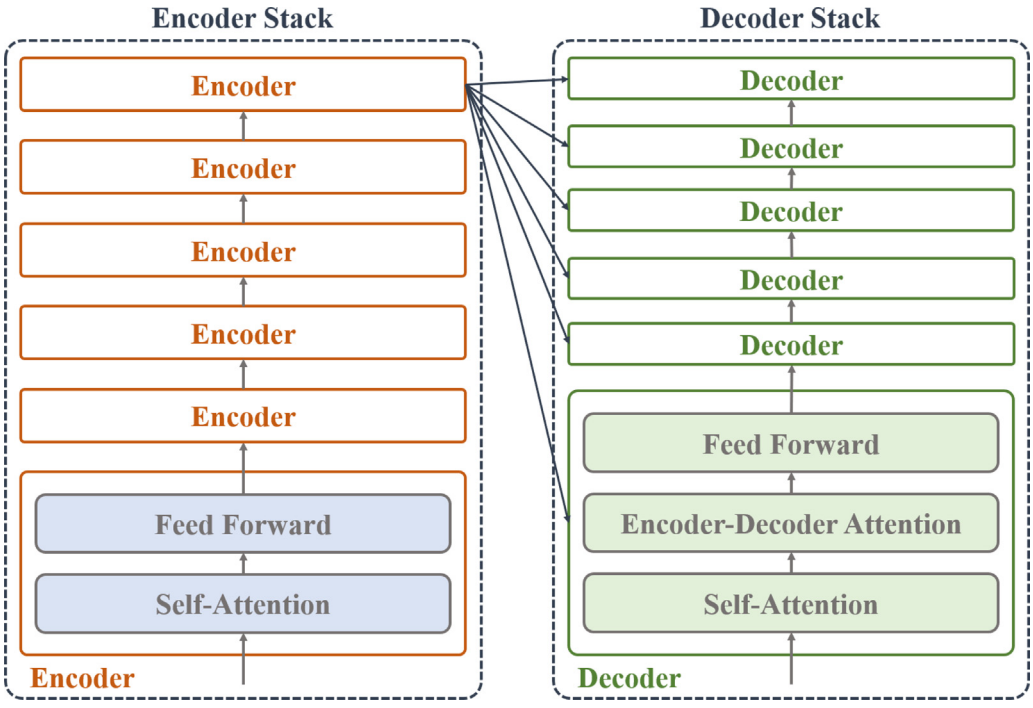


Fig. 2-2. Architecture of Transformer's encoder and decoder [5].

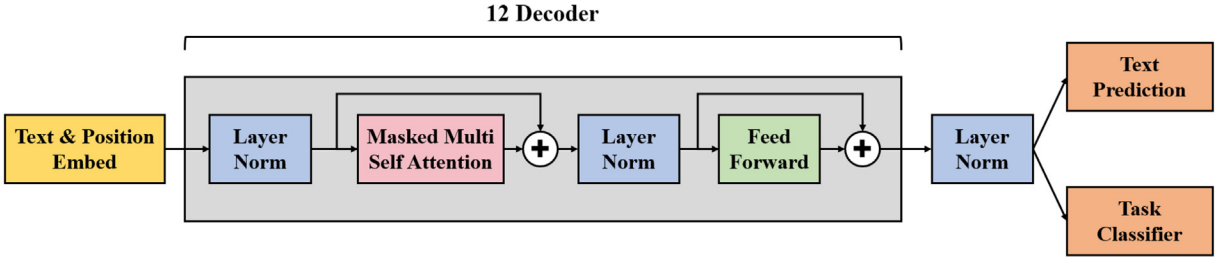


Fig. 2-3. Typical GPT-2 architecture [7].

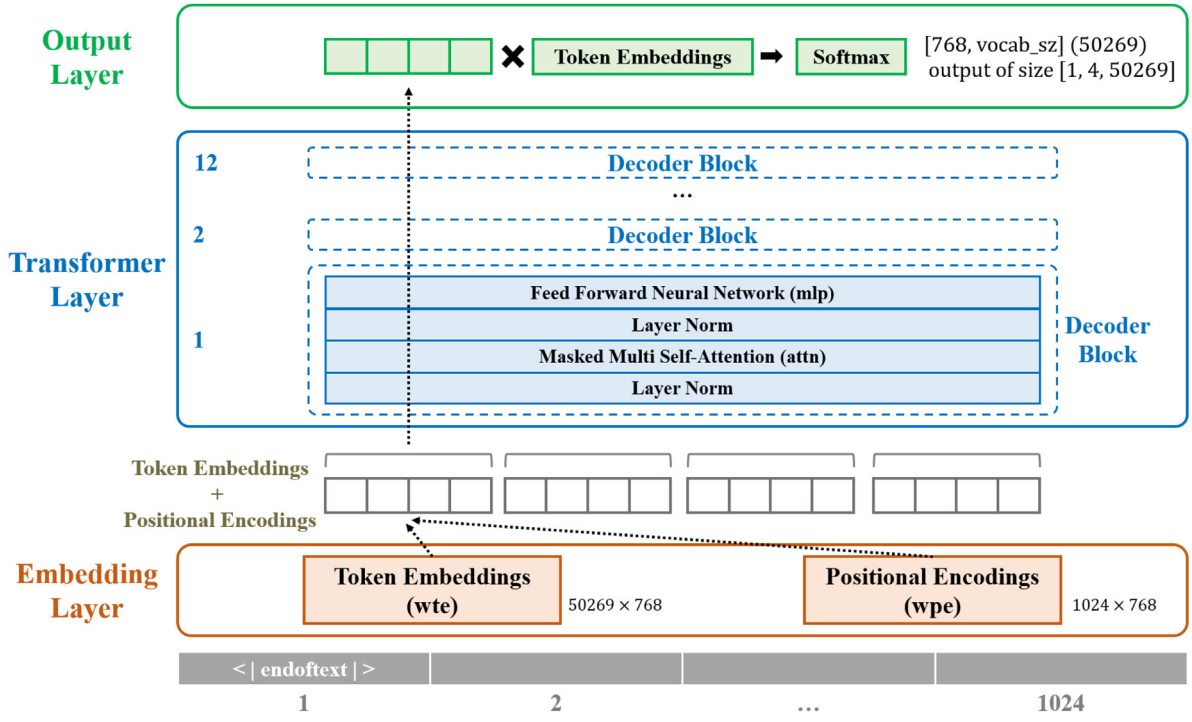


Fig. 2-4. Proposed conditional generative pretrained Transformer-2 [7].

minimum dimensions, hence embedding dimension = 768 for each token. GPT-2 has no sense of the beginning or end of a document within a larger text when fine-tuning. Thus, GPT-2 trains models by separating texts using the single-token < | endoftext | >, which is required to properly handle quotes and line breaks in each lyrics text document. Before pushing the embeddings representation to the decoder block, we use a suffixing marker in the embedding matrix to illustrate relationships between words. Since GPT-2 is not an recurrent neural network, we added a positional embedding layer to give the model a specific sense of position for each word (wpe), as shown in Fig. 2-5, right. If the input context size has 1024 tokens, position encoding dimension = [1024, 768].

The trained GPT-2 model contains two weight matrices: token embeddings and position encoding. Each location has a position encoding, with the position encoding and token embeddings treated as a new embedding to perform the original self-attention operation, as shown in Fig. 2-6. Once we have the new embedding, it passes through each of the 12 GPT-2 layers, where each layer is a transformer decoder block comprising two sublayers, including attention and feedforward networks.

The transformer layer is a stack of decoders. The new embedding enters the multiple repeated decoder block with the following structure.

1. Masked multi-head attention (attn).
   This contains two Conv1D layers, Attention Dropouts (*attn_pdrop*) (rate = 0.1) and Residual Dropouts (*resid_pdrop*) (rate = 0.1). It deals with information about the various connections between words. Total computational cost is similar to that for a full dimensional single-head section attention due to the reduced number of dimensions per head.
2. Add & norm.
   We add masked multi-head attention input and output to obtain another sequence, which is subsequently layer normalized, i.e., values for each data point's different dimensions are adjusted to mean = 0 and standard deviation = 1. This study set Layer Norm size = 768 and epsilon = 1e-05.
3. Feedforward (mlp).
   The vector goes through two linear transformation layers, or more precisely two linear layers. We use the GELU activation function in the feedforward network and add Dropouts (0.1).
4. Add & Norm.
   Joins the same Layer Norm as discussed above.

These 4 structures construct a decoder, and the Transformer layer has 12 identical decoders. Similar to the output layer, we pass inputs to a final Layer Norm (768 dimensions) and through a Linear layer with a final dimension [768, vocab_size](50269) to obtain output size [1, 4, 50269]. This output represents the next input word and we pass this through a Softmax layer to obtain the word position inside the vocabulary with the highest probability. After many repetitions we obtain the probability distribution for desired output words. Arguments [8] used for training the lyrics transfer generation model are shown in Table 2-2.

Bena et al. [9] proposed a unique language generation system that could produce creative poetry verses, using a fine-tuned GPT-2 adaptation. They also considered generating poetry in various languages using transfer learning, and trained 8 models for 8 different emotions, each on a sub-corpus predominantly demonstrating a particular emotion. This paper attempts to create art in the form of auto-generated poetry, while opening the door to more text based tasks involving emotional elicitation and influenced creative neural generation.

2.3. StanfordNLP

StanfordNLP [10] offers a range of tools to process human natural language and simplify text analysis, collated from research by the StanfordNLP Group, including part of speech tagger, named entity recognizer, dependency parsing, etc. StanfordNLP is built using highly accurate neural networks, allowing efficient training and evaluation using your own marker data, and these modules are based on the PyTorch architecture.

This study only uses the dependency parsing StanfordNLP tool to capture relationships between words. Hence we only discuss that tool. A dependency parser [11] is a tool to extract a dependency parse of a sentence, representing the grammatical structure and relationships between "head" words and words. A dependency parse can viewed as a

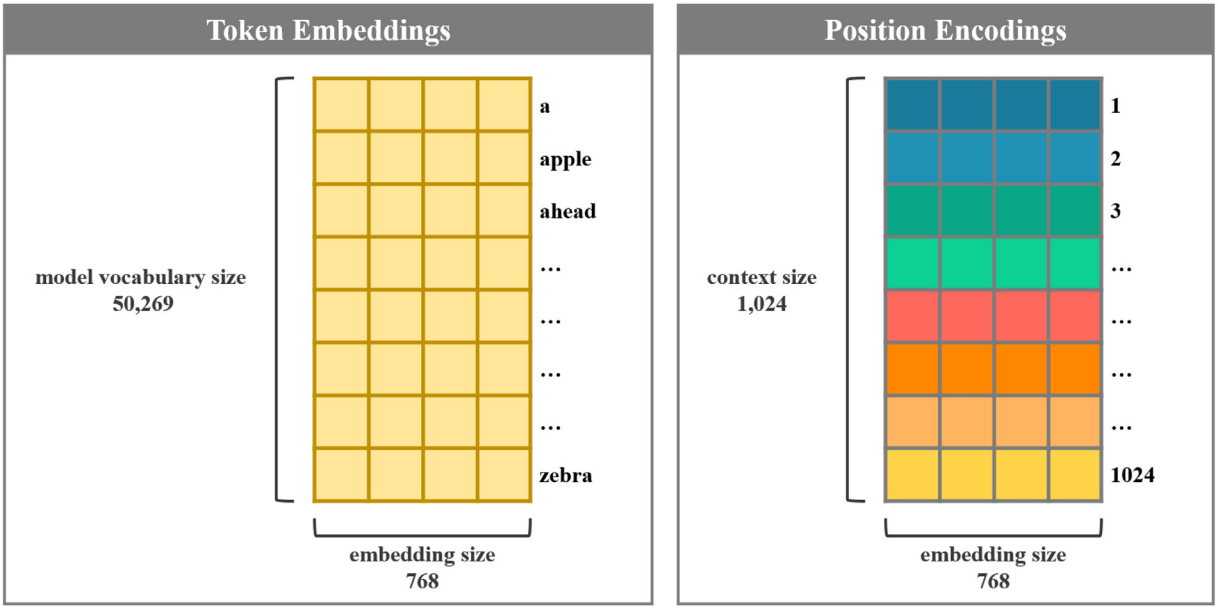


Fig. 2-5. Token embeddings and position encodings of proposed model [7].

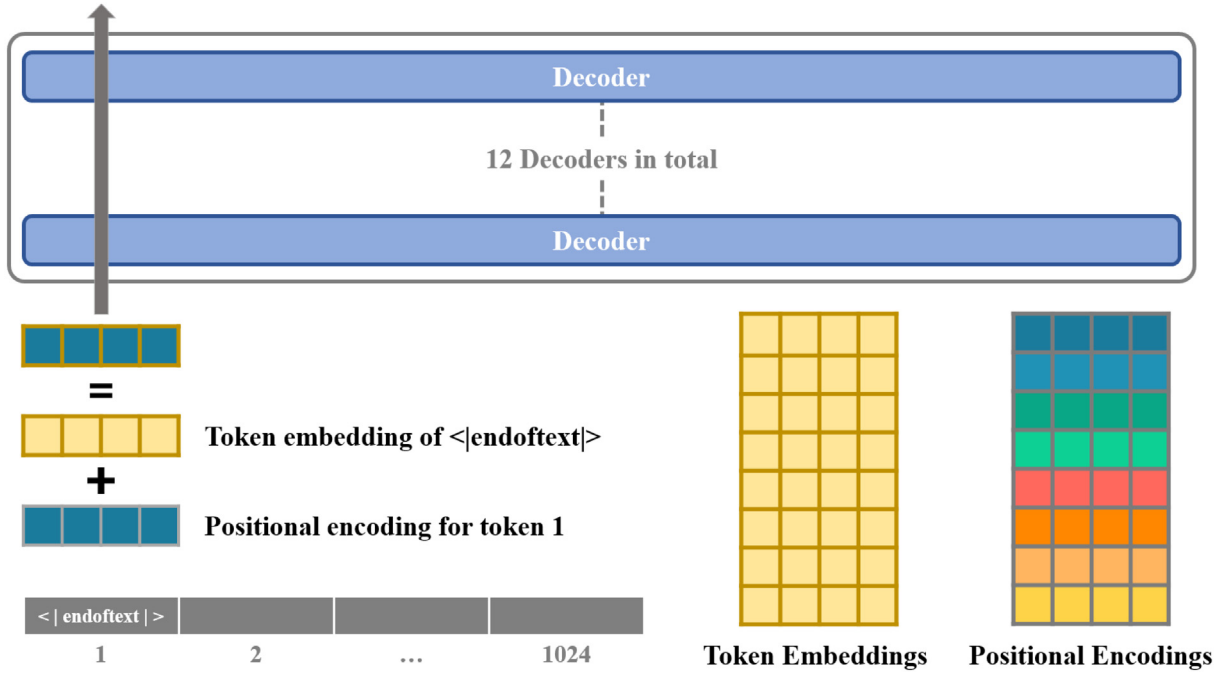


Fig. 2-6. Combined token embedding and position encoding for the proposed GPT-2 model [7].

Table 2-2
Argument table for the proposed lyrics transfer generation model.

Parameter	Definition	Value
n_head	Number of attention heads for each attention layer in the Transformer encoder	12
n_layer	Number of hidden layers in the Transformer encoder	12
n_ctx	Dimensionality of the causal mask	1024
n_embd	Dimensionality of embeddings and hidden states	768
n_positions	Maximum sequence length that this model could be used with	1024
layer_norm_epsilon	epsilon to use in layer normalization layers	1e−05
vocab_size	Vocabulary size of the GPT-2 model	50 269
embd_pdrop	Dropout ratio for embeddings	0.1
attn_pdrop	Dropout ratio for attention	0.1
resid_pdrop	Dropout probability for all fully connected layers in embeddings, encoder, and pooler	0.1

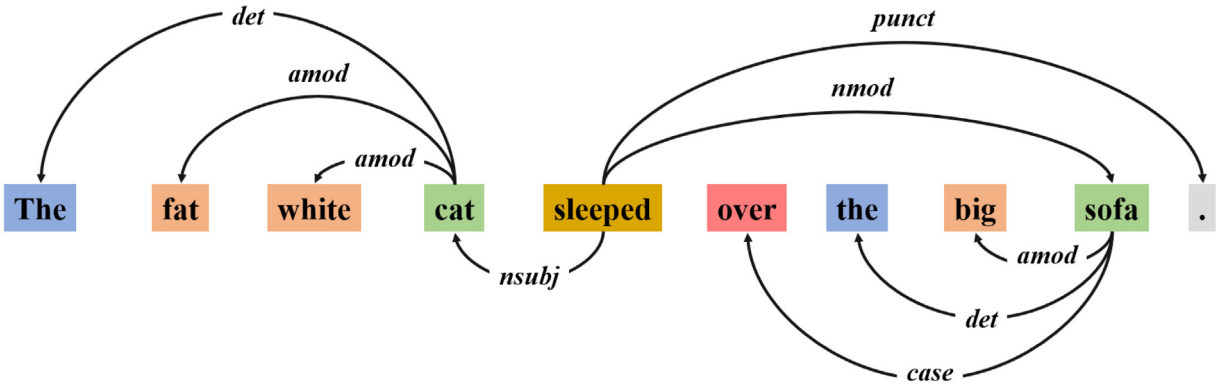


Fig. 2-7. Example dependency parser.

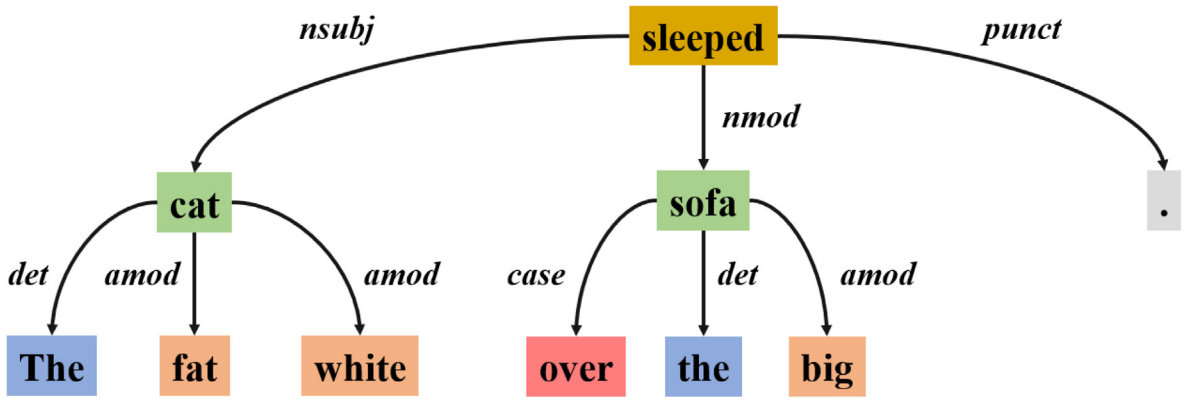


Fig. 2-8. Example expanded representation of standard dependencies.

tree, where each node is a word in the sentence and edges correspond to relations between the words. Fig. 2-7 shows a typical dependency parser example. The arrow from "cat" to "white" indicates that "white" modifies "cat", and the label assigned to the arrow (*advmod*) describes the particular dependency.

The current dependency parser contains approximately 50 grammatical relations [12]. Dependencies are all binary relations between a governor (or head) and dependent word. The Stanford Dependency representation includes several variants that suit different goals. One is a more basic representation of Fig. 2-7, which is useful for direct dependency parsing. The expanded representation adds additional relations that cannot be expressed in a tree structure but may be useful to capture semantic relationships between entities in a sentence. Fig. 2-8 shows the expanded representation for standard dependencies using the sentence from Fig. 2-7

For example, "Amy plays the piano every day." and " Did Jacky play the violin yesterday?" are with a similar structure. With the dependency parser of StanfordNLP, we can find the word pairs "(play, piano)" and "(play, violin)" belonging to the same dependency relation. Therefore, we can capture such word pairs for replacement and can make sure the grammatical structure generally correct. This study applies this concept by replacing the word pairs that have a similar structure to the original lyrics.

2.4. Text style transfer generation literature review

Text style transfer rephrases a given sentence into a different style, which is an important function for a wide range of NLP applications. Generally, style transfer can be divided into two parts: supervised style transfer with parallel data, and unsupervised with non-parallel data. Recent advances in text generation have used large amounts of parallel data, such as machine translation and abstraction. Carlson et al. [13] used aligned parallel text for prose style transfer. Bibles provide such a corpus, with their well-demarcated sentence and verse structure making the corpus equally useful for various other natural language tasks. This paper used a multilayer RNN encoder and multilayer RNN with attention for decoding, creating a style transfer specific architecture. Jhamtani et al. [14] used parallel data for text style transfer, transforming Modern English text to Shakespearean style English by enriching a seq2seq model. The model used a bidirectional long term short memory model to encode the input modern English sentence, and the decoder combined an RNN and pointer network module. Pretraining embeddings for words related to tackle limited the amount of parallel data. Fig. 2-9 shows the overall architecture.

Transferring languages from one style to another has been previously trained using parallel data. However, despite the various successes, dictionary constraints can cause sparse problems and there is very limited parallel data available for different styles. This has prompted recent interest in style transfer without a parallel corpus for text generation problems. Gero et al. [15] proposed a simple AWD-LSTM neural language model architecture and pretrained weights for style-specific text generation. They divided the data into three style datasets and split style data into sentences for training examples. Most unsupervised methods split text content and style [16].

Hu et al. [17] built a neural generative model with VAE and holistic attribute discriminators or effectively generate sentences with controllable attributes by learning latent representations. Fig. 2-10 shows how sentence $\hat{x}$ is generated conditioned on latent z, where c contains each attribute used to set a separate discriminator to measure how well the generated sample matches the desired attribute, which also improves generator results. Blue and red arrows in the figure indicate model independence judgment and model optimization methods, respectively.

Gao et al. [18] proposed a multitask learning approach incorporating a Seq2Seq module as the encoder and an autoencoder (AE) module.

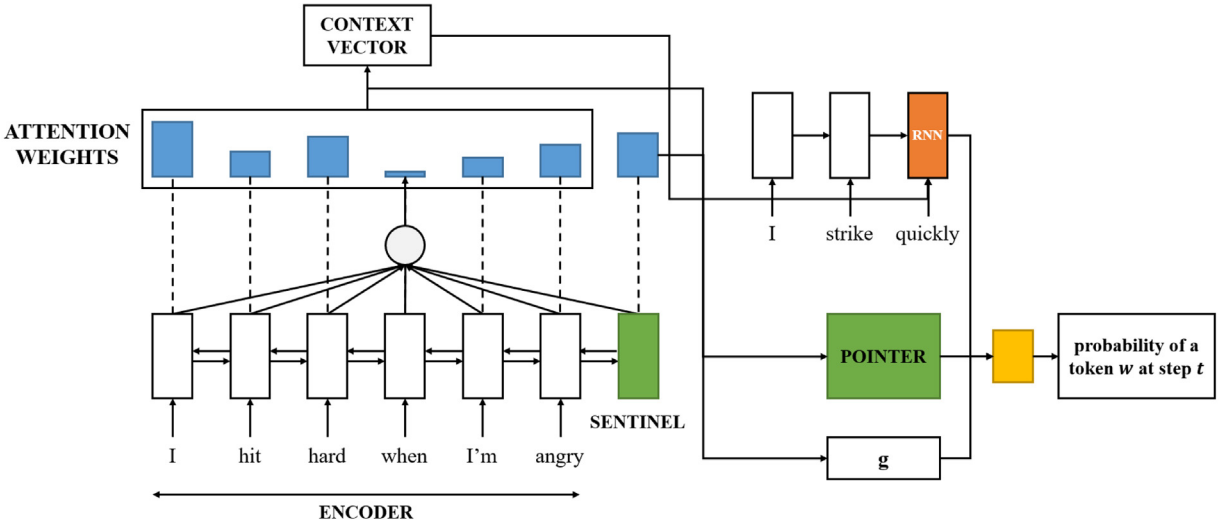


Fig. 2-9. Overall architecture for Jhamtani et al. [14].

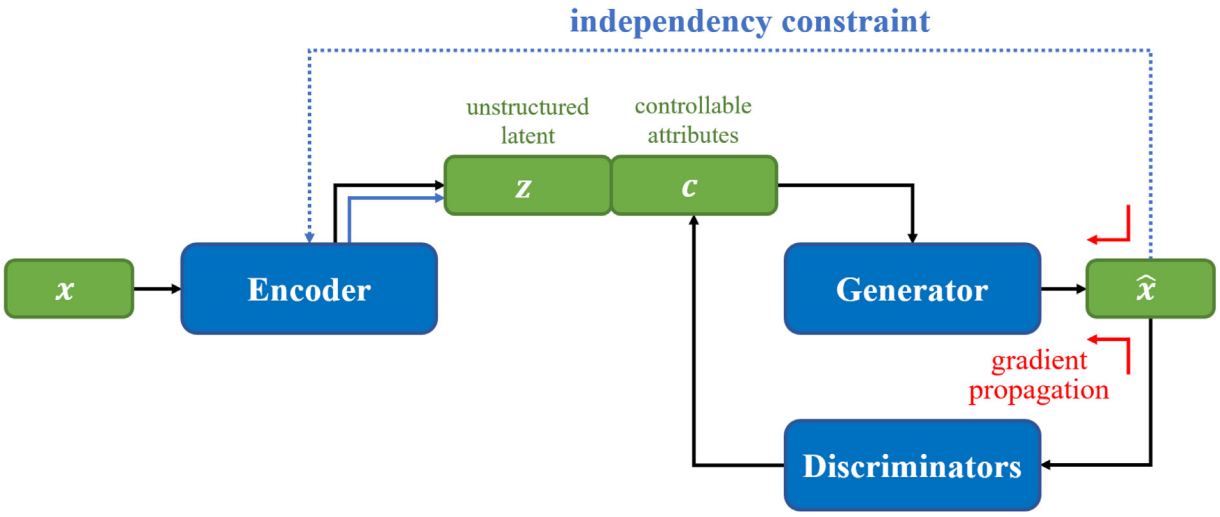


Fig. 2-10. Generative model proposed by Hu et al. [17].

Both share the same decoder, as shown in Fig. 2-11. The Seq2Seq module produces the prediction representation, then AE obtains a representation for the corresponding responses and stylized sentences. This structure allows non-parallel style transfer by sharing latent space. Experiments using the proposed approach for generating responses of the targeted style outperformed baseline.

Many proposed methods borrow concepts from generative adversarial network (GAN) frameworks for generative adversarial training discriminators. Jin et al. [19] proposed a stylistic headline generation to enrich headlines with three style options (humor, romance, and clickbait) for improved attraction. Jin et al. proposed a multitask framework that adopts Seq2Seq based on Transformer architecture to summarize styles and introduces style-guided encoder attention into the multi-head attention module. They also proposed a parameter sharing scheme to enhance summarizing and stylizing capabilities. Fig. 2-12 shows the proposed model architecture.

Syed et al. [20] proposed an author stylized rewriting language model (StyleLM) with two parts: unsupervised pretraining with a Transformer based language model on a large English dataset cascaded them into an encoder–decoder like framework; and author specific fine-tuning with denoising Auto-Encoder loss (DAE loss), allowing the decoder to push the target author's style while rewriting the encoder input text, as shown in Fig. 2-13.

3. Methodology

3.1. System architecture of lyrics style transfer generation

This study demonstrates the effectiveness of the GPT-2 model in generating stylized lyrics. Most previous studies on style transfer generation of lyrics do not consider creativity and audio fit simultaneously; whereas the proposed lyrics style transfer generation includes a GPT-2 language model, dependency parser module, and rhyme modification module. The goal was to generate diverse and appropriate sentences based on given style. Experiments were conducted on an English music dataset containing pop and rock music.

Fig. 3-1 shows the overall architecture for the proposed method, with more detailed description in subsequent sections. Section 3.2 discusses training and fine-tuning the model and Section 3.3 describes optimizing the controlled generation lyrics. Section 3.4 proposes the lyrics style transfer generation algorithm.

3.2. Model training and fine-tuning

The proposed model was trained on a lyrics dataset collected by scraping outbound links on the Genius lyrics website, which is the world's largest collection of song lyrics and crowdsourced musical

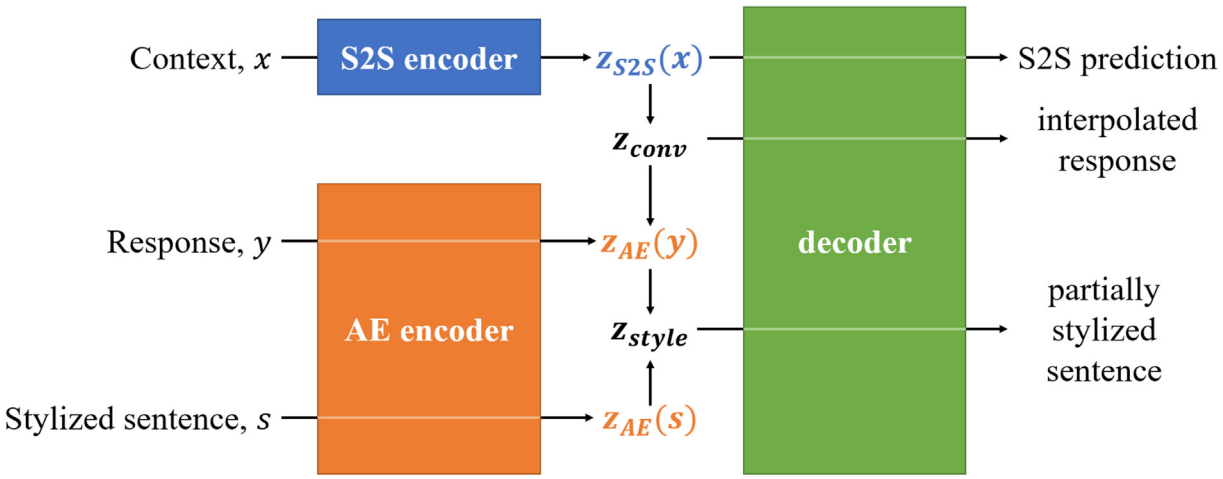


Fig. 2-11. Model architecture proposed by Gao et al. [18].

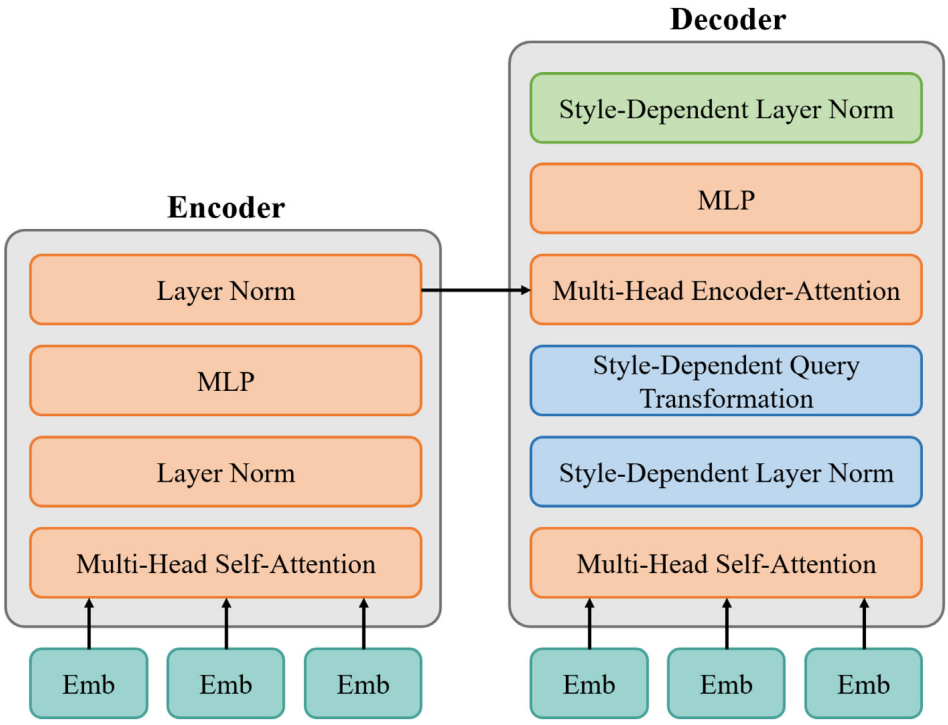


Fig. 2-12. Model architecture proposed by Jin et al. [19].

knowledge. This study collected lyrics from five genres: pop, country, rock, rap, and reggae; with the songs collected spanning different years. Each style has a unique approach to lyric writing and the vocabulary of the lyric texts varies. The dataset included 2024, 1204, 2159, 2299, and 266 songs in each genre, respectively. Each song included information regarding The artist, year of publication, album, title, and lyrics.

At its core, GPT-2 learns English language structure by observing billions of word, sentence, and paragraph examples, scraped from all corners of the internet. Therefore, it learns good English language models. For example, the word “’black”’ can be used to describe a person, an object, etc., and the correct meaning can be clearly identified by the trained model. Transfer learning eliminates the need to re-train from scratch by inheriting pretrained models from previously learned models in other areas. Thus, GPT-2 can train a model for style-specific lyrics generation under several conditions by utilizing transfer learning and building upon OpenAI’s GPT-2 text generation model. Basically, we take a pretrained model and train it with a specific song style lyrics dataset, leading the model towards generating the specific lyric style. Fig. 3-2 shows the process to re-train GPT-2.

We first coded the lyrics text and then retrained the GPT-2 model with the objective: predict the next word given all of the previous words within some text, i.e., similar to ’auto-regression. First we uploaded a file containing the lyrics to be used to tune the model. For example, suppose pop song lyrics were used as training data, then and all the data is combined into a single text, splitting each lyric with <| endoftext| >.

We retrained GPT-2 for several reasons. The different lyric styles use different expressions and lyricism, e.g. pop lyrics tend to emphasize romantic love, whereas rock lyrics tend to emphasize social or political aspects. The different styles often use different words, e.g. pop songs usually contain “love”, “feel”, “live”, “heart”, and “tell”; whereas rock songs usually contain “oh”, “never”, “burn”, and “rock”. We had some limitations song lyric dataset size, so we focused on pop and rock styles, combining pop and rock lyric data to form a text file with over 8000

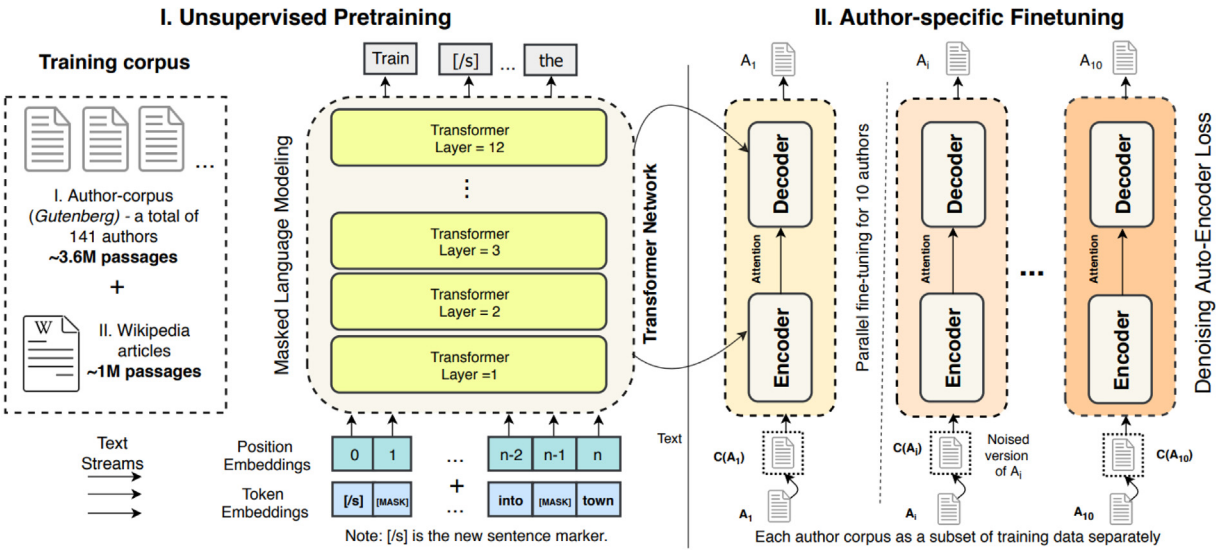


Fig. 2-13. The model architecture of Syed et al. [20].

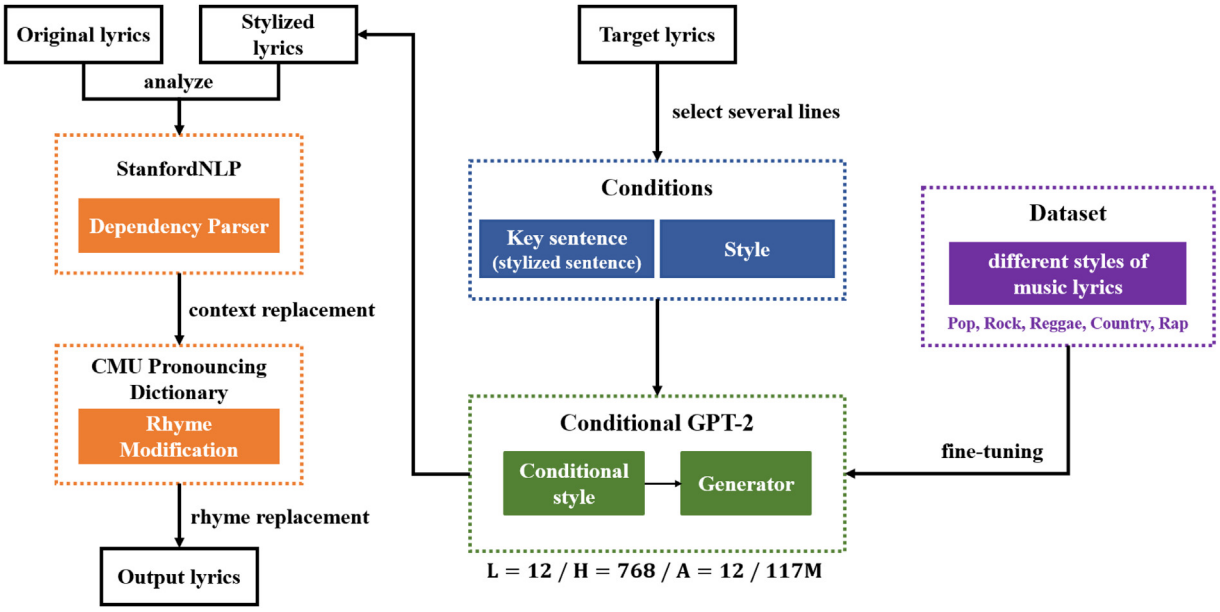


Fig. 3-1. Proposed lyrics style transfer generation framework.

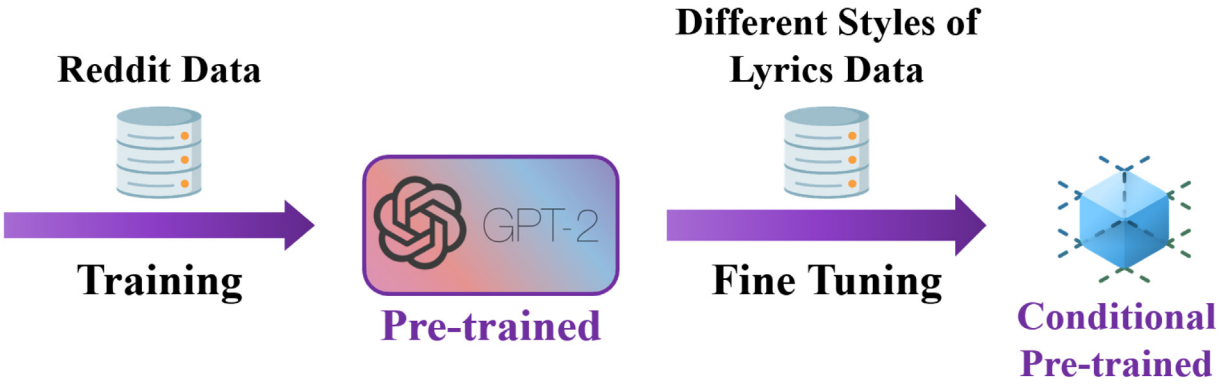


Fig. 3-2. Process to re-train GPT-2.

lines as the training dataset. The next step was to encode the dataset, encoding generates the file.

We used the pretrained GPT-2 small model (117M) from the Transformer library. The pretrained model was fine-tuned using a lyric

Table 3-1
Key dependencies used in this study.

Abbreviation	Name	Sentence	Representation
nsubj	Nominal subject	Tim defeated Amy	*nsubj* (defeated, Tim)
nsubjpass	Passive nominal subject	Amy was defeated by Tim	*nsubjpass* (defeated, Amy)
amod	Adjectival modifier	Sam eats red meat	*amod* (meat, red)
nmod	Noun modifier	Filled up with water	*nmod* (filled, water)
advmod	Adverb modifier	Genetically modified food	*advmod* (modified, genetically)
dobj	Direct object	She gave me a raise	*dobj* (gave, raise)
compund	Compound nouns	Wait at this bus stop	*compound* (bus, stop)

dataset including many unique songs accompanied by metadata, i.e., genre, artist, year, album, and song title. As discussed above GPT-2 output was then used to generate new lyric text content based on style-specific lyric features. For example, the lyrics could be constrained to be like rock style, with the key sentence "I've seen your shadow in the dark I've seen this struggle in your life" as a prefix. The generate function forces the text to start with the given character sequence and generate lyrics from there. Transfer learning for the GPT-2 language model enables creating good transfer-generated models in the desired lyric style. The motivation behind this is that differences between pretrained models with different conditions are expected to reflect lyric differences as well as musical styles.

3.3. Optimizing controlled generation lyrics

Two modules were designed to tune the lyrics and allow other orientations to be considered in the style transformation rather than just generating text: dependency parser (DP) and rhyme modification (RM). Thus, two model variants were generated, GPT-2+DP and GPT-2+DP+RM.

The DP module provides a representation of the grammatical relationships between words in a sentence and was designed to be easily understood and effectively used by those who want to extract textual relationships. GPT-2 simply generates lyrics based on a given style, regardless of whether the lyric structure is consistent with the original lyrics. This study considers the original song structure and uses the StanfordNLP tool to adjust the GPT-2 generated lyrics using a three part approach. First we analyze the GPT-2 generate lyrics for dependency relationships using DP. There were approximately 50 dependency types with seven selected as the key dependencies for this study, KD = {*nsubj, nsubjpass, amod, nmod, advmod, dobj, compund*}, as shown in Table 3-1. Each lyric may include several dependencies, but only dependencies in KD were recorded. After analyzing the dependencies, the lyrics are replaced with the corresponding relationship positions, and each lyric line is replaced similarly.

Song lyrics generally have a rhythm to the lines and usually include rhymes. While not essential, rhyming is a key element for singability and musical compatibility. A rhyme is repetition of the same sound, usually at the end of each line, and helps emphasize the rhythm. To further improve rhyming fluency for the proposed model, we implemented a post-processing RM step using the Carnegie Mellon University (CMU) pronunciation dictionary (also known as CMUdict) [21], an open-source pronunciation dictionary originally developed by the CMU Speech Group for speech recognition research. The rhyme modification module was designed to consider the rhyme from the original lyrics and the lyrics' contextual semantics. First, rhyme analysis is performed on the original and stylized lyrics modified by DP, producing a word list with the same rhyme for each rhyme in the original lyrics. Then the stylized lyrics are used to calculate similarity for each word in the list, and the one with the highest similarity is used as the replacement. Thus, the overall module design moves towards lyrics generation with style control and rhythmic structure.

3.4. Lyrics style transfer generation algorithm

GPT-2 has shown promising results generating text for creative uses, and text style transfer can transform input text from one style to another. Therefore, the proposed approach addresses creative text generation, producing stylized lyrics that consider style and key sentence. The model outputs one token at a time, similar to traditional language models. This output token can be added at the end of the input token, and this new sequence then be used as input to generate the next token. This study differs from traditional approaches in that it does not aim to change the style of a given text, but rather, to design a module for generating lyrics with controllability.

This work proposes a language generative model framework that can control the output text style using a style reference example. The generator takes a sentence from a source style as input and transfers it to the target style. Generative algorithms are used to tell the model style patterns and embedded content, and particularly characteristics for the different styles. Scarce parallel data for many style transfer tasks has prompted interest in style transfer without a parallel corpus [16]. This study captured pop, rap, country, rock, and reggae lyric styles from the Genius website [22], which is non-parallel data, to overcome sparse data problems when generating different lyric styles. Each style included songs from various artists and excluded songs with "remix", "live", or "radio edit" in their title to prevent duplication. Each song included metadata for genre, artist, year, album, and song title. The top 60 songs were then captured by sorting them according to artist's song popularity. Each style was fine-tuned to the pretrained model using the lyric dataset, and trained according to the original lyrics. Thus, pop style and rock style models were generated by fine-tuning GPT-2. GPT-2 can be thought of as a stack of decoders. Each row of the embedding matrix corresponds to the embedding vector for a word in the model's vocabulary, hence the result obtained by multiplication operation is the corresponding score for each word in the vocabulary.

We used a nucleus sampling mechanism [23] to select words with a high probability, i.e., using the probability distribution to determine the sample set, rather than directly specifying a fixed number. Three parameters were defined as follows

- *top_k*: only k tokens with the highest probability are kept;
- *top_p*: retain the cumulative probability ≥ *top_p*; and
- *logits*: the shape of logits distribution, the product of batch size and vocabulary size.

For example, suppose we set $top\_p = 0.95$, $logits = 5$ (i.e., batch size = 1 and vocabulary size = 5), with probabilities $[0.62, 0.17, 0.15, 0.03, 0.03]$, and hence cumulative probability = $[0.62, 0.79, 0.94, 0.97, 1]$. Higher score means higher probability the word will become the next word. Since $0.94 < 0.95$ and $0.97 > 0.95$, we choose the first three tokens. The model outputs a word each iteration, and continues to iterate until a complete sequence is generated (defined as output lyrics Y). Generation is stopped when sequence length = 1024 or when a terminator is generated in the sequence. Fig. 3-3 shows the token output process.

Suppose we have two lyric text datasets $X = \{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$ (original lyrics) and $Y = \{y^{(1)}, y^{(2)}, \ldots, y^{(n)}\}$ (output lyrics after GPT-2 processing), with styles S_x and S_y, respectively (e.g. S_x is pop style and S_y is rock style). The dataset is non-parallel, i.e., data does not contain pairs $(x^{(m)}, y^{(n)})$ that describe the same content. The style transfer goal

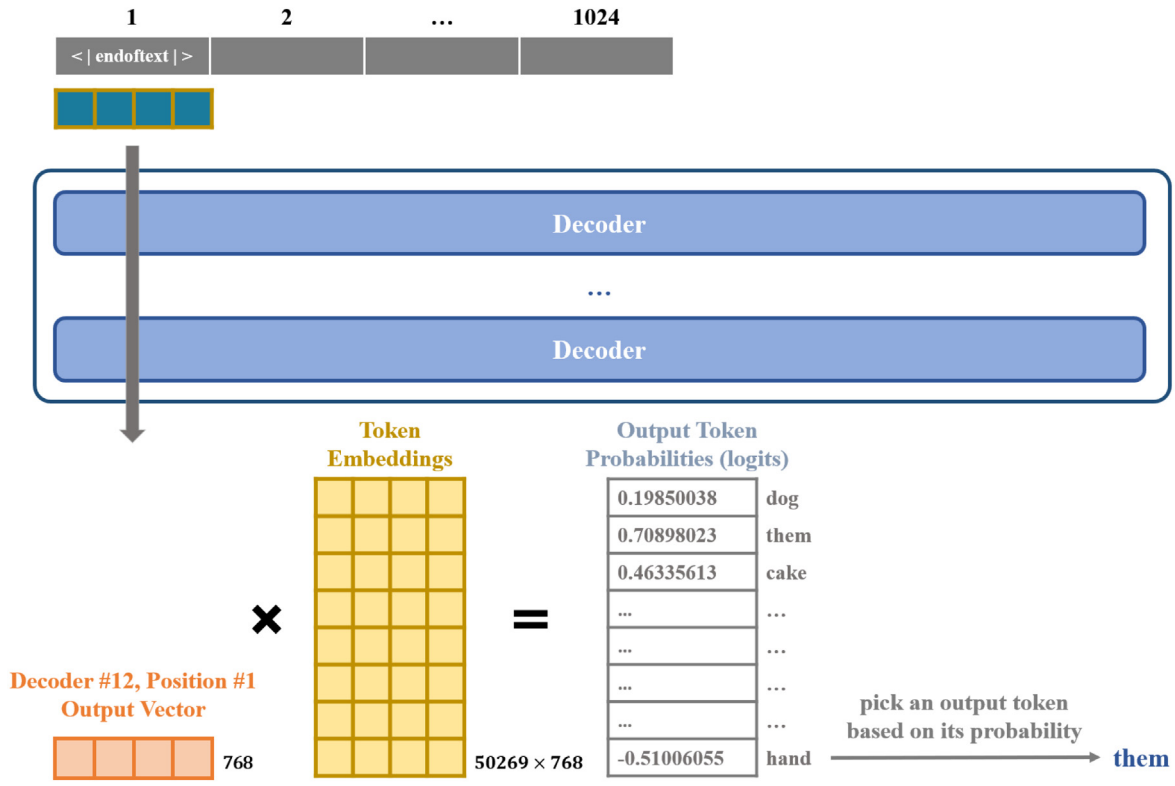


Fig. 3-3. Proposed token output process.

Table 3-2
Lyrics style transfer notations used in this study.

Notations	Description
X	Lyrics original text
Y	Lyrics target text
S_x	Lyrics original style
S_y	Lyrics target style
K	Key sentence
G	The generation result from GPT-2
M	The modification result from StanfordNLP

is to transfer data x with style S_x to style S_y and vice versa. Table 3-2 defines the notation used in this study.

Drop-down menus were designed for the user to adjust the lyrics for the style they want to generate. Two adjustable parameters were provided: key lyric start line and lyric style genre. However, the generative model space becomes more restricted as more features are set by the user, e.g. if the key phrase is too long, the output lyrics may appear less diverse with poorer quality.

Suppose pop music style lyrics are transferred to rock music. Then X (structure template) is the provided pop music lyrics and Y is the target rock music lyrics. A few lines from the target lyrics are chosen as the key sentence K so GPT-2 can include rock style elements in its generation (G). GPT-2 can set random seeds as control parameters for generation to ensure G will not be the same as X. For this study, we randomly chose an integer random seed ∈ [0,100] for each generation, to ensure the proposed model can achieve a creative goal. Batch size was used to specify how many lyrics versions the model generates, with later lyrics generally performing better.

Previous GPT-2 lyric generation approaches only considered the text, ignoring original lyric structure. The next step in the proposed approach was to post-processed the output lyrics using DP and RM, as discussed in Section 3.3, producing modified lyrics M, using the dependencies defined by StanfordNLP to form a tree structure.

Table 3-3
Original lyric's dependencies.

Word 1	Dependency	Word 2
('say', 'VBP')	*nsubj*	('They', 'PRP')
('say', 'VBP')	*ccomp*	('girl', 'NN')
('girl', 'NN')	*nsubj*	('Taylor', 'NNP')
('girl', 'NN')	*cop*	('was', 'VBD')
('girl', 'NN')	*det*	('a', 'DT')
('girl', 'NN')	*amod*	('good', 'JJ')

Table 3-4
Dependency results for StanfordNLP replacement.

Word 1	Dependency	Word 2
('make', 'VBP')	*nsubj*	('You', 'PRP')
('make', 'VBP')	*ccomp*	('sea', 'NN')
('sea', 'NN')	*nsubj*	('river', 'NN')
('sea', 'NN')	*cop*	('was', 'VBD')
('sea', 'NN')	*det*	('a', 'DT')
('sea', 'NN')	*amod*	('black', 'JJ')

Table 3-3 shows an example of basic dependencies for the line from the original song $x^{(m)}$ = "They say Taylor was a good girl". Each line dependency is divided into three parts: each word in the sentence on the right, word correspondences on the left, and dependencies between words in the middle. We must analyze the dependencies in the same way as the original lyrics and replace them with synonyms with the same relationship. Table 3-4 shows the lyrics change to "You make river was a black sea", which has same dependencies between words as in the original lyrics and adds a rock element to the sentence.

Finally, we can obtain lyrics with specific stylistic and structural similarities to the original song. Section 4 shows that the proposed method effectively transfers the style to the lyrics and generates satisfying lyrical content. We then apply RM post-processing optimization and iterates over each line of the original lyrics, replacing the closing words from the DP treated lyrics with words that match the rhythm

Fig. 4-1. Example lyrics results.

and contextual meaning of the original lyrics. One of the original lyrics ends with "girl", so the lyrics change from "You make river was a black sea" to "You make river was a black pearl".

4. Experiments

4.1. Dataset

Song lyrics are widely available across the internet in the form of user-generated content. Two large-scale datasets are used in our experiments. Two large-scale datasets are used in this experiment. The first dataset is a large corpus of 7952 English lyrics crawled from the Genius lyrics website and used to fine-tune the lyric style conversion generation model. The second dataset is the audio of the original lyrics, and the DALI dataset [24] was chosen for this study, which is a large dataset of audio full sections synchronized with audio, lyrics and notes, with lyrics and notes (of the vocal melody) aligned in time. Due to its data integrity, the dataset for this study is composed of 5358 songs with English lyrics, each of which contains audio, lyrics and midi notes.

4.2. Evaluation metrics

In general, it is difficult to judge the quality of lyrics generated automatically by a computer. Therefore, we used both automatic and human evaluation to validate our system in terms of feasibility and usability of our proposed method for lyrics transfer generation. For the same input context but different reference examples of the same style, our framework should generate different output texts but all with the same style. Aligning with prior work proposed by Nikolov et al. [25], this study used the automatic metrics of overlap score. The overlap score tests our model's ability to generate original and unique content words, and it ranges from 0 to 1, with 0 indicating high originality and 1 indicating low originality. The lower the overlap score, the more original the generated lyrics are. The overlap score is defined as shown in Eq. (1), where x is the input lyric text and y is the generated output lyrics text.

$$overlap(x, y) = \frac{|\ \{y\} \cap \{x\}\ |}{\{y\}} \tag{1}$$

Due to the lyrics generation is more like human creation, the human evaluation should be a better way for performance evaluation. The human evaluation can be a more comprehensive measurement. Therefore, we invited the evaluators with good English education and passion for English songs to participate in the human evaluation for the proposed conditional GPT-2 generation method compared with GPT-2, GPT-2+DP, and GPT-2+DP+RM. We chose a sample of 100 stylized lyrics texts generated by the proposed system for each transfer task for users to cross-rate, with 50 each being pop to rock and rock to pop conversion, respectively. Table 4-1 shows the five metrics used to evaluate the final lyrics, including thematic (T), structural (S), originality (O), meaningfulness (M) and Fitness (F) scores; and Figs. 4-1 and 4-2 show the interface for the web based survey for lyrics transfer generation. All participants were asked to rate the lyrics on a scale from 1 to 5, based on a five-point Likert scale. Here "1" means completely disagree, and "5" means completely agree. The higher the score, the better the performance.

4.3. Experimental evaluation

Figs. 4-3 to 4-8 show example lyrics transferred using the proposed and comparison methods, with the original lyrics on the left and stylized lyrics on the right. Fig. 4-3 and Fig. 4-6 show stylized lyrics using GPT-2 from pop to rock and rock to pop, respectively. We set the key sentence as "Bury every word I've said in the city of the dead and drown this masterpiece in red" (rock style) and conditionally transferred from pop to rock style. The generated stylized lyrics include elements from the key sentence and the generated sentences are meaningful and

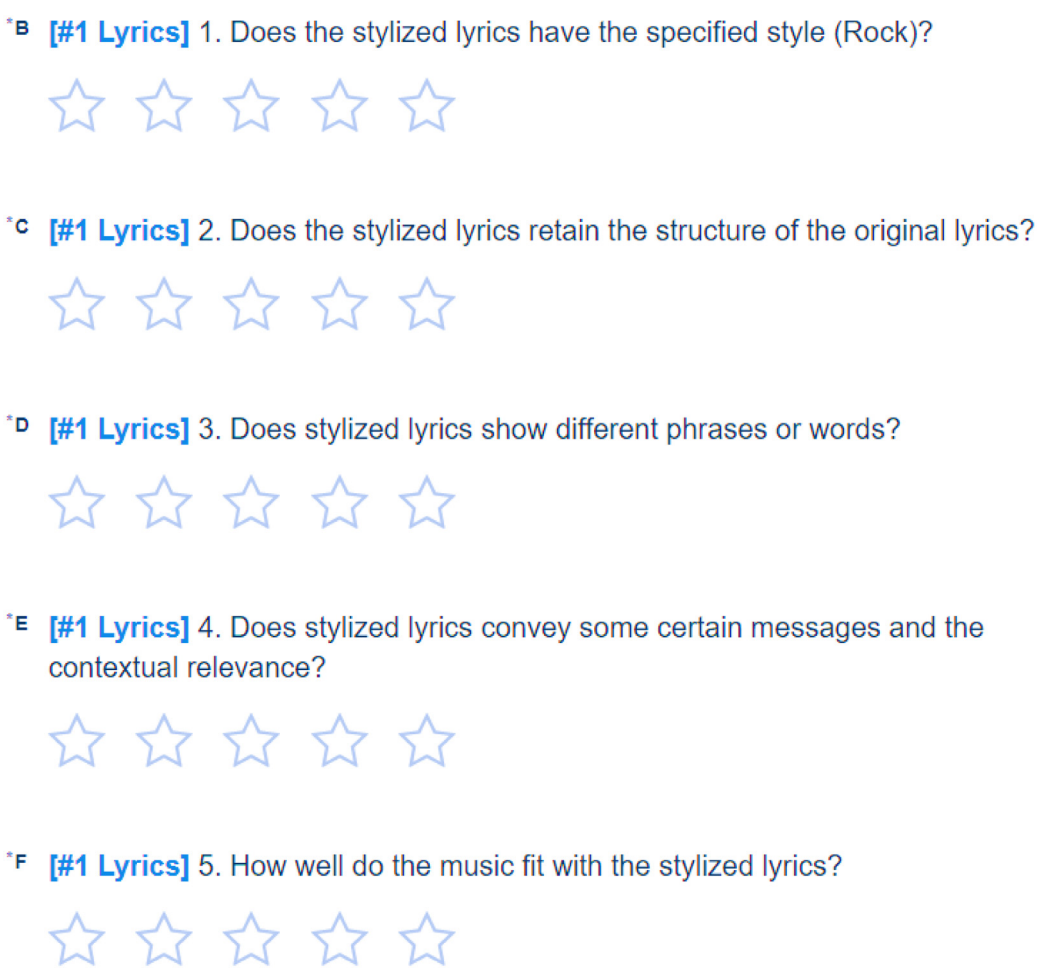


Fig. 4-2. Web based performance metrics questionnaire.

Table 4-1
Five metrics for human evaluation.

Metric	Question	Reference
Thematic (T)	Does the stylized lyrics have the specified style?	Yang et al. [26]
Structural (S)	Does the stylized lyrics retain the structure of the original lyrics?	Watanabe et al. [27]
Originality (O)	Does stylized lyrics show different phrases or words?	Lu et al. [28]
Meaningfulness (M)	Does stylized lyrics convey some certain messages and the contextual relevance?	Oliveira [29]
Fitness (F)	How well do the music fit with the stylized lyrics?	Oliveira [29]

Table 4-2
Results of overlap score.

Method	Style transfer from rock (Pop2Rock)	Style transfer from pop (Rock2Pop)
	Average of 50 tracks (Mean ± Std)	Average of 50 tracks (Mean ± Std)
GPT-2	0.2893 ± 0.0674	0.3039 ± 0.0702
GPT-2 + DP	0.6683 ± 0.1351	0.6962 ± 0.0996
GPT-2 + DP + RM	0.5500 ± 0.1137	0.5798 ± 0.0774

grammatically correct. Key words "bury", "dead", and "red" appear in key sentences, and similar concepts are mentioned in the generated results, e.g. "bury the bones", "search of flesh", "death or holy ghosts", etc. Figs. 4-4 and 4-7 show that post-processing with DP. We used DP to find out the candidates of word pairs to replace and modify the original lyrics. Figs. 4-5 and 4-8 show that further postprocessing with RM ensures end of sentence rhymes that match the original lyrics.

Human and automatic evaluation are described in Section 4.2. We first report the results of the automatic evaluation of our proposed system. Table 4-2 show the results for overlap scores, including means and standard deviations. The generated results are divided into two parts, one is Pop to Rock (Pop2Rock) and the other is Rock to Pop (Rock2Pop). A total of 100 songs in the experimental section, 50 Pop to Rock and 50 Rock to Pop, will be evaluated for each of the three different approaches. The experimental results show that GPT-2 yields 0.2893 ± 0.0674 in Pop2Rock and 0.3039 ± 0.0702 in Rock2Pop, which is the lowest overlap value among all methods, indicating that the GPT-2 model is capable of generating original song texts. In Pop2Rock, GPT-2+DP has an overlap of 0.6683 ± 0.1351, and in Rock2Pop, GPT-2+DP has an overlap of 0.6962 ± 0.0996, which is even higher than GPT-2 alone because GPT-2+DP draws on the structure of the original pop or rock lyrics, thus increasing the overlap score, but the overlap is within acceptable limits and the content still retains the results produced by GPT-2. It can be learned that the dependency substitution method designed in this study can capture important relations and make substitutions for similar terms, but not for less important relations, so the overlap rate is higher than the former. In other words, GPT-2+DP combines stylized text with the structure of the original lyrics. GPT-2+DP+RM was modified for each concluding phrase of the GPT-2+DP results, yielding 0.5500 ± 0.1137 in Pop2Rock and 0.5798 ± 0.0774 in Rock2Pop, with overlap scores even lower than GPT-2+DP, indicating that some rhyming words can be effectively substituted.

# 1 Original Lyrics (Pop)	# 1 Stylized Lyrics (GPT-2 / Rock)
They say Taylor was a good girl Never one to be late Complain, express Ideas in her brain Working on the night shift Passing out the tickets You're gonna have to pay her If you wanna park here Well, mommy's little dancer Has quite a little secret Working on the streets now Never gonna keep it It's quite an imposition And now she's only wishin' That she would have listened To the words they said Poor Taylor She just wanders around Unaffected by The winter winds, yeah And she'll pretend that Well she's somewhere else So far and clear About two thousand miles From here Well, Peter Patrick pitter-patters on the window The sunny silhouette won't let him in Poor old Pete's got nothing, cause he's been falling Somehow, Sonny knows just where he's been He thinks that singing on Sunday is gonna save his soul Now that Saturday's gone Sometimes he thinks That he's on his way But I can see That his brake lights are on He just wanders around Unaffected by The winter winds, yeah And he'll pretend that Well he's somewhere else So far and clear About two thousand miles From here Such a tough enchilada Filled up with nada Giving what she gotta Give to get a dollar bill Used to be a limber chicken Times have been a tickin' Now she's finger lickin' To the man With the money in his pocket Flying in his rocket Only stopping by On his way to a better world If Taylor finds a better world Then Taylor's gonna run away	You don't need sand in your mountains, rivers, rivers to bury your bones They'll stone you to death or holy ghosts to see you fly Your hands'll make you dirtier than before But if you're gonna swim the river deep and white, then Maybe your river's gonna carry you, maybe your river's gonna carry you With every breath you make a sudden drop, a wailing sound That pulls you to the deep black sea in search of flesh Awake in the morning and find you're drowning in your own tears 'Cause you don't need sand in your mountains, rivers, rivers to bury your bones Their pleasure-seeking partner behind you is under your sea And unless you go to the city in a boat or car, what you find there Is a hole in your river bed, what you find there They'll stone you to death or holy ghosts to see you fly And your hands'll make you dirtier than before But if you're gonna swim the river deep and white, then Maybe your river's gonna carry you, maybe your river's gonna carry you With every breath you make a sudden drop, a wailing sound That pulls you to the deep black sea in search of flesh Awake in the morning and find you're drowning in own tears 'Cause you don't need sand in your mountains, rivers, rivers to bury your bones River Awake in the morning and find you're drowning In your own tears Oh you don't need sand in your mountains river, rivers, rivers to bury your bones River Awake in the morning and find you're drowning In your own tears Oh you don't need sand in your mountains river, rivers, rivers to bury your bones To bury your bones

Fig. 4-3. The result of the GPT-2 processing of the lyrics from Pop to Rock.

# 1 Original Lyrics (Pop)	# 1 Stylized Lyrics (GPT-2 + DP / Rock)
They say Taylor was a good girl	You make river was a black sea
Never one to be late	Never one to be late
Complain, express	Complain, express
Ideas in her brain	Awake in her morning
Working on the night shift	Pulls on the night sea
Passing out the tickets	Find out the drowning
You're gonna have to pay her	You're gonna carry to bury bones
If you wanna park here	If you fly what before
Well, mommy's little dancer	That pulls black sea
Has quite a little secret	Has before a sudden drop
Working on the streets now	Carry on the streets maybe
Never gonna keep it	Never you find drowning
It's quite an imposition	Bury quite an bones
And now she's only wishin'	And now river's sudden drop
That she would have listened	That you would have dirtier
To the words they said	To the words you find
Poor Taylor	Find tears
She just wanders around	Gonna carry before around
Unaffected by	Make breath
The winter winds, yeah	The winter you go
And she'll pretend that	And she'll bury bones
Well she's somewhere else	Well she's somewhere else
So far and clear	So far and clear
About two thousand miles	About two thousand miles
From here	From here
Well, Peter Patrick pitter-patters on the window	Well, Peter Patrick search on the flesh
The sunny silhouette won't let him in	The sunny silhouette you dirtier bones in
Poor old Pete's got nothing, cause he's been falling	Deep sea own got nothing, cause he's been falling
Somehow, Sonny knows just where he's been	Somehow, you go then maybe carry been
He thinks that singing on Sunday is gonna save his soul	River deep that drop sound on Sunday is gonna save his soul
Now that Saturday's gone	Now that hands'll make
Sometimes he thinks	Sometimes he thinks
That he's on his way	You fly on his way
But I can see	But gonna carry can see
That his brake lights are on	That his own tears are drop
He just wanders around	You maybe don't around
Unaffected by	Make breath
The winter winds, yeah	The winter there hole
And he'll pretend that	And he'll find drowning
Well he's somewhere else	Well he's somewhere else
So far and clear	So far and clear
About two thousand miles	About two thousand miles
From here	From here
Such a tough enchilada	Such a black sea
Filled up with nada	Pulls up with sea
Giving what she gotta	Given you drowning find
Give to get a dollar bill	Give to bury a dollar bones
Used to be a limber chicken	Used to be a limber chicken
Times have been a tickin'	Partner have been a sea
Now she's finger lickin'	Before dirtier finger you
To the man	To the man
With the money in his pocket	With the hole in his bed
Flying in his rocket	Search in his flesh
Only stopping by	Maybe carry by
On his way to a better world	On his way to be a deep sea
If Taylor finds a better world	If gonna carry black sea
Then Taylor's gonna run away	There find you run away

Fig. 4-4. The result of the GPT-2+DP processing of the lyrics from Pop to Rock.

# 1 Original Lyrics (Pop)	# 1 Stylized Lyrics (GPT-2 + DP + RM / Rock)
They say Taylor was a good girl	You make river was a black pearl
Never one to be late	Never one to be late
Complain, express	Complain, express
Ideas in her brain	Awake in her again
Working on the night shift	Pulls on the night drift
Passing out the tickets	Find out the crickets
You're gonna have to pay her	You're gonna carry to bury fur
If you wanna park here	If you fly what disappear
Well, mommy's little dancer	That pulls black lancer
Has quite a little secret	Has before a sudden top-secret
Working on the streets now	Carry on the streets pshaw
Never gonna keep it	Never you find unfit
It's quite an imposition	Bury quite an dentition
And now she's only wishin'	And now river's sudden wishin'
That she would have listened	That you would have rechristened
To the words they said	To the words you read
Poor Taylor	Find bailor
She just wanders around	Gonna carry before around
Unaffected by	Make sigh
The winter winds, yeah	The winter you pah
And she'll pretend that	And she'll bury fat
Well she's somewhere else	Well she's somewhere else
So far and clear	So far and clear
About two thousand miles	About two thousand miles
From here	From here
Well, Peter Patrick pitter-patters on the window	Well, Peter Patrick search on the lindo
The sunny silhouette won't let him in	The sunny silhouette you dirtier bones in
Poor old Pete's got nothing, cause he's been falling	Deep sea own got nothing, cause he's been falling
Somehow, Sonny knows just where he's been	Somehow, you go then maybe carry been
He thinks that singing on Sunday is gonna save his soul	River deep that drop sound on Sunday is gonna save his soul
Now that Saturday's gone	Now that hands'll on
Sometimes he thinks	Sometimes he thinks
That he's on his way	You fly on his way
But I can see	But gonna carry can see
That his brake lights are on	That his own tears are spawn
He just wanders around	You maybe don't around
Unaffected by	Make sigh
The winter winds, yeah	The winter there pah
And he'll pretend that	And he'll find vat
Well he's somewhere else	Well he's somewhere else
So far and clear	So far and clear
About two thousand miles	About two thousand miles
From here	From here
Such a tough enchilada	Such a black armada
Filled up with nada	Pulls up with casada
Giving what she gotta	Given you drowning ricotta
Give to get a dollar bill	Give to bury a dollar quill
Used to be a limber chicken	Used to be a limber chicken
Times have been a tickin'	Partner have been a tickin'
Now she's finger lickin'	Before dirtier finger lickin'
To the man	To the man
With the money in his pocket	With the hole in his socket
Flying in his rocket	Search in his pocket
Only stopping by	Maybe carry by
On his way to a better world	On his way to be a deep curled
If Taylor finds a better world	If gonna carry black curled
Then Taylor's gonna run away	There find you run away

Fig. 4-5. The result of the GPT-2+DP+RM processing of the lyrics from Pop to Rock.

Fig. 4-6. The result of the GPT-2 processing of the lyrics from Rock to Pop.

The results of the human evaluation are shown in Tables 4-3, 4-4, Figs. 4-9 and 4-10. The results of the GPT-2 model are first described, where the performance is better in terms of thematic and originality, and worse in terms of structure and applicability. Thus, the GPT-2 model can generate lyrical content that is very different from the original lyrics and retains the specified style conditions. Several studies have also shown that Transformer outperforms traditional models such as LSTM in feature extraction because GPT-2 uses a large amount of high quality pre-trained data, allowing the model to be applied to text generation with good results. In addition, the pre-trained GPT-2 model is fine-tuned with different style text features. Therefore, from the results, it can be confirmed that the proposed GPT-2 generation model has an advantage in the style transfer of lyrics.

However, GPT-2 is less structured and less fitness because it does not take into account additional conditions, such as the analysis of the structure of the original lyrics. From the GPT-2 + DP results, it was found that the crowd's satisfaction with the structure and fitness of the song was increased compared to GPT-2, mainly due to the structure with reference to the original lyrics. However, there is a slight decrease in originality compared to GPT-2, presumably limited by structure and dependencies, so it is not possible to replace all the words. Overall, the current results are within the acceptable range. In the GPT-2+DP+RM model, structure and fitness received the highest scores of all methods. However, the meaningfulness is not as good as GPT-2, because the post-processing process causes a situation where the meaning of the lines in the substituted lyrics may be unclear, and "one leak sinks a great ship" occurs. In terms of originality, the results are no better than the GPT-2 method, because in the post-processing stage, there are more restrictions on the substitution of lyrics, both in terms of rhyme and linguistic similarity. Clearly, the results of the overlap scores showed the same trend as the indicators of originality for human evaluation, with both being best for GPT-2 originality. Indeed, the fact that our model quickly obtains a huge performance comes from our fine-tuning strategy on a pre-trained model.

Fig. 4-7. The result of the GPT-2+DP processing of the lyrics from Rock to Pop.

Table 4-3
Results of human evaluation (Pop2Rock).

Model	Metrics	Rater			Mean ± Std
		1	2	3	
GPT-2	Thematic (T)	4.4	4.5	3.9	4.2667 ± 0.2625
	Structural (S)	1.6	1.9	1.9	1.8000 ± 0.1414
	Originality (O)	4.5	4.0	4.1	4.2000 ± 0.2160
	Meaningfulness (M)	3.1	3.0	3.8	3.3000 ± 0.3559
	Fitness (F)	1.4	1.9	1.8	1.7000 ± 0.2160
GPT-2+DP	Thematic (T)	3.5	4.0	4.0	3.8333 ± 0.2357
	Structural (S)	3.9	4.1	3.8	3.9333 ± 0.1247
	Originality (O)	3.0	3.5	3.6	3.3667 ± 0.2625
	Meaningfulness (M)	3.2	3.7	3.2	3.3667 ± 0.2357
	Fitness (F)	3.1	3.0	3.5	3.2000 ± 0.2160
GPT-2+DP+RM	Thematic (T)	3.8	3.6	3.9	3.7667 ± 0.1247
	Structural (S)	4.3	4.4	3.7	4.1333 ± 0.3091
	Originality (O)	3.5	3.5	3.2	3.4000 ± 0.1414
	Meaningfulness (M)	2.9	3.4	3.4	3.2333 ± 0.2357
	Fitness (F)	4.7	3.8	3.8	4.1000 ± 0.4243

Table 4-4
Results of human evaluation (Rock2Pop).

Model	Metrics	Rater			Mean ± Std
		1	2	3	
GPT-2	Thematic (T)	4.3	4.5	4.4	4.4000 ± 0.0816
	Structural (S)	1.7	1.5	2.3	1.8333 ± 0.3399
	Originality (O)	4.1	3.9	4.3	4.1000 ± 0.1633
	Meaningfulness (M)	3.4	3.6	3.9	3.6333 ± 0.2055
	Fitness (F)	1.5	1.2	1.7	1.4667 ± 0.2055
GPT-2+DP	Thematic (T)	3.6	3.7	4.4	3.9000 ± 0.3559
	Structural (S)	3.6	3.9	3.5	3.6667 ± 0.1700
	Originality (O)	3.5	3.1	4.5	3.7000 ± 0.5888
	Meaningfulness (M)	3.7	3.8	3.0	3.5000 ± 0.3559
	Fitness (F)	3.5	3.2	3.4	3.3667 ± 0.1247
GPT-2+DP+RM	Thematic (T)	4.1	3.5	4.5	4.0333 ± 0.4110
	Structural (S)	4.4	3.8	4.3	4.1667 ± 0.2625
	Originality (O)	3.8	3.6	3.6	3.6667 ± 0.0943
	Meaningfulness (M)	3.4	3.3	3.5	3.4000 ± 0.0816
	Fitness (F)	4.1	4.1	3.9	4.0333 ± 0.0943

# 3 Original Lyrics (Rock)	# 3 Stylized Lyrics (GPT-2 + DP + RM / Pop)
All that we can do is just survive	All that we can do is too arrive
All that we can do to help ourselves is stay alive	All that we can do to help he is said survive
All that we can do is just survive	All that we can do is just dive
All that we can do to help ourselves is stay alive	All that we can do to help he is said survive
Ragged lines of ragged grey	sweet home of ragged 'kay
Skeletons, they shuffle away	Skeletons, they bit hey
Shouting guards and smoking guns	shake hand and smoking ones
Will cut down the unlucky ones	Will cut down the black guns
I clutch the wire fence until my fingers bleed	I clutch the wire fence until my I disagreed
A wound that will not heal	A wound he will not feel
A heart that cannot feel	A heart that I'd deal
Hoping that the horror will recede	Hoping that the He will skied
Hoping that tomorrow we'll all be freed	Hoping that tomorrow knew when be decreed
Sickness to insanity	locked to vanity
Prayer to profanity	Prayer to vanity
Days and weeks and months go by	Days and weeks and months locked bonsai
Don't feel the hunger, too weak to cry	Don't feel the hunger, out smoke to sigh
I hear the sound of gunfire at the prison gate	I hitched the sound of gunfire at the prison crate
Are the liberators here	Are the when we're
Do I hope or do I fear?	Do I when or do I we're
For my father and my brother, it's too late	For my father and my brother, it's too wait
But I must help my mother stand up straight	But I must didn't my mother stand up eight
Are we the last ones left alive?	Are we the last ones black live
Are we the only human beings to survive?	Are we the only hualive ride to alive
Are we the last ones left alive?	Are we the black smoke left survive?
Are we the only human beings to survive?	Are we the only human door to live
I hear the sound of gunfire at the prison gate	I locked the sound of gunfire at the prison grate
Are the liberators here	Are the when we're
Do I hope or do I fear?	Do ride hitched or do ride domineer?
For my father and my brother, it's too late	For my father and my brother, it's when ate
But I must help my mother stand up straight	But I must knew my mother stand up late
Are we the last ones left alive?	Are we the last ones sweet live
Are we the only human beings to survive?	Are we the only human looked to alive
Are we the last ones left alive?	Are we the tight tree left survive?
Are we the only human beings to survive?	Are we the only human enough to strive

Fig. 4-8. The result of the GPT-2+DP+RM processing of the lyrics from Rock to Pop.

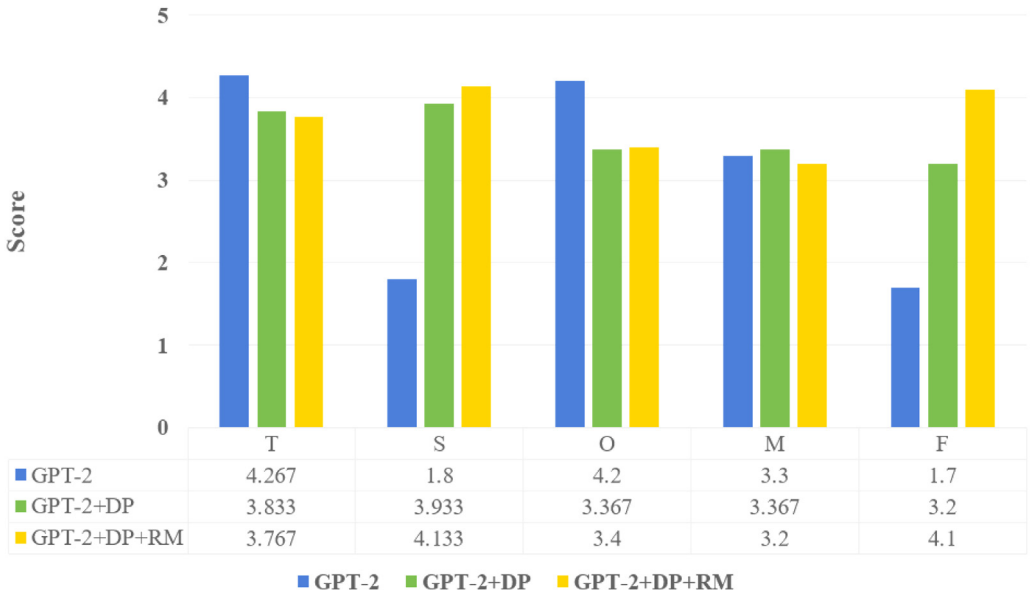
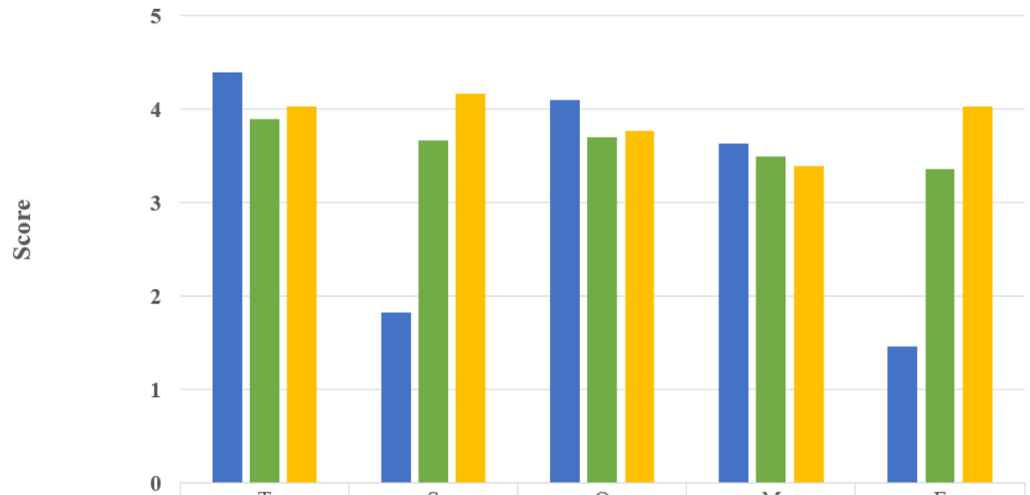


Fig. 4-9. Results of human evaluation (Pop2Rock).



	T	S	O	M	F
GPT-2	4.4	1.833	4.1	3.633	1.467
GPT-2+DP	3.9	3.667	3.7	3.5	3.367
GPT-2+DP+RM	4.033	4.167	3.767	3.4	4.033

Fig. 4-10. Results of human evaluation (Rock2Pop).

5. Conclusions

The main contributions of this study are as the following three points. First, we use the GPT-2 model to learn the deep representations of different music styles and to generate the corresponding context according to the given style. Second, we use the dependency parser tool of StanfordNLP to select the word pairs from the generated result of GPT-2. And then, we use the word pairs to modify an original lyric of the given style for maintaining the whole structure of the lyric and fitting the context of the given style. Third, we develop the rhyme modification module to modify the end words of each sentence in the lyric for fitting the singability of the songs.

This study is based on the GPT-2 model, which generates different styles of lyrics under specific conditions (key sentences and genre style). In addition, the structure and rhythm of the lyrics are considered. The proposed method not only can generate original textual content, but music and lyrics also be matched. In short, the proposed model achieves the lyrics generation consistent structure and rhythm according to the given music style. In the end, we conducted experiments with two evaluation metrics, including automatic and human evaluation. The experimental results demonstrate that the proposed method generates lyrics of the given style and outperforms competitive baselines.

CRediT authorship contribution statement

Jia-Wei Chang: Conceptualization, Investigation, Methodology. **Jason C. Hung:** Supervision, Validation. **Kuan-Cheng Lin:** Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the Ministry of Science and Technology, Taiwan, R.O.C. [grant number MOST 108-2218-E-025-002-MY3], [grant number MOST 109-2221-E-025-009]. Special thanks to Miss Ching-Yi Chiou for her assistance in the development of the programming for this study.

References

[1] R. Barzilay, K.R. Mckeown, Information Fusion for Multidocument Summerization: Paraphrasing and Generation (Ph.D. thesis), Columbia University, 2003.

[2] A. Prakash, S.A. Hasan, K. Lee, V. Datla, A. Qadir, J. Liu, O. Farri, Neural paraphrase generation with stacked residual LSTM networks, in: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, 2016, pp. 2923–2934, arXiv:abs/1610.03098.

[3] Y. Zhao, P. Yu, S. Mahapatra, Q. Su, C. Chen, Discretized bottleneck in VAE: Posterior-collapse-free sequence-to-sequence learning, 2020, arXiv preprint arXiv:2004.10603, December.

[4] S. Golovanov, R. Kurbanov, S. Nikolenko, K. Truskovskyi, A. Tselousov, T. Wolf, Large-scale transfer learning for natural language generation, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 6053–6058, July.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, et al., Attention is all you need, in: Advances in Neural Information Processing Systems, NIPS, 2017, pp. 5998–6008.

[6] Paweł Budzianowski, Ivan Vulić, Hello, it's GPT-2 – How can I help you? Towards the use of pretrained language models for task-oriented dialogue systems, in: Proceedings of the 3rd Workshop on Neural Generation and Translation, WNGT 2019, 2019, pp. 15–22, November.

[7] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, Language Models are Unsupervised Multitask Learners, OpenAI blog, 2019.

[8] Huggingface.co., Openai GPT2 — Transformers 3.0.2 documentation, 2020, Retrieved from https://huggingface.co/transformers/model_doc/gpt2.html.

[9] B. Bena, J. Kalita, Introducing aspects of creativity in automatic poetry generation, 2020, arXiv preprint arXiv:2002.02511.

[10] C.D. Manning, M. Surdeanu, J. Bauer, J.R. Finkel, S. Bethard, D. McClosky, The Stanford CoreNLP natural language processing toolkit, in: Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, 2014, pp. 55–60, June.

[11] Peng Qi, Timothy Dozat, Yuhao Zhang, Christopher D. Manning, Universal Dependency parsing from scratch, in: Proceedings of the {CoNLL} 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, 2018, pp. 160–170, October.

[12] M.C. De Marneffe, C.D. Manning, Stanford Typed Dependencies Manual, Technical Report, Stanford University, 2008, pp. 338–345.

[13] K. Carlson, A. Riddell, D. Rockmore, Evaluating prose style transfer with the Bible, R. Soc. Open Sci. 5 (10) (2018) 171920.

[14] H. Jhamtani, V. Gangal, E.H. Hovy, E. Nyberg, Shakespearizing modern language using copy-enriched sequence-to-sequence models, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, 2017, September.

[15] K.I. Gero, G. Karamanolakis, L. Chilton, Transfer learning for style-specific text generation, in: Proceedings of 32nd Conference on Neural Information Processing Systems, NIPS 2018, 2018.

[16] H. Gong, S. Bhat, L. Wu, J. Xiong, W.M. Hwu, Reinforcement learning based text style transfer without parallel training corpus, in: Proceedings of the 17th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, 2019, pp. 3168–3180.
[17] Z. Hu, Z. Yang, X. Liang, R. Salakhutdinov, E.P. Xing, Toward controlled generation of text, in: Proceedings of the 34th International Conference on Machine Learning, ICLR, 2017, pp. 1587–1596.
[18] X. Gao, Y. Zhang, S. Lee, M. Galley, C. Brockett, J. Gao, B. Dolan, Structuring latent spaces for stylized response generation, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP) and the 9th International Joint Conference on Natural Language Processing, IJCNLP, 2019, pp. 1814–1823.
[19] D. Jin, Z. Jin, J.T. Zhou, L. Orii, P. Szolovits, Hooks in the headline: Learning to generate headlines with controlled styles, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL, 2020, pp. 5082–5093.
[20] B. Syed, G. Verma, B.V. Srinivasan, A. Natarajan, V. Varma, Adapting language models for non-parallel author-stylized rewriting, in: Proceedings of the 34th Conference on Artificial Intelligence, AAAI, 2020, pp. 9008–9015, February.
[21] CMUdict, 2014, Retrived from https://github.com/cmusphinx/cmudict.
[22] Song Lyrics & Knowledge, Genius. Retrieved from https://genius.com/.
[23] A. Holtzman, J. Buys, L. Du, M. Forbes, Y. Choi, The curious case of neural text degeneration, in: Proceedings of International Conference on Learning Representations, ICLR, 2019.
[24] G. Meseguer-Brocal, A. Cohen-Hadria, G. Peeters, Dali: A large dataset of synchronized audio, lyrics and notes, automatically created using teacher-student machine learning paradigm, in: Proceedings of 19th International Society for Music Information Retrieval Conference, ISMIR, 2018, September.
[25] N.I. Nikolov, E. Malmi, C.G. Northcutt, L. Parisi, Conditional rap lyrics generation with denoising autoencoders, 2020, arXiv:abs/2004.03965.
[26] X. Yang, X. Lin, S. Suo, M. Li, Generating thematic chinese poetry using conditional variational autoencoders with hybrid decoders, in: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI, 2018, pp. 4539–4545.
[27] K. Watanabe, Y. Matsubayashi, S. Fukayama, M. Goto, K. Inui, T. Nakano, A melody-conditioned lyrics language model, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Vol. 1, 2018, pp. 163–172, June.
[28] X. Lu, J. Wang, B. Zhuang, S. Wang, J. Xiao, A syllable-structured, contextually-based conditionally generation of chinese lyrics, in: Proceedings of the 16th Pacific Rim International Conference on Artificial Intelligence: Trends in Artificial Intelligence, PRICAI 2019, in: Lecture Notes in Computer Science, vol. 11672, Springer, Cham, 2019, pp. 257–265.
[29] H.G. Oliveira, Tra-la-lyrics 2.0: Automatic generation of song lyrics on a semantic domain, J. Artif. Gen. Intell. 6 (2015) 87–110.