

Received 27 July 2022, accepted 22 August 2022, date of publication 26 August 2022, date of current version 9 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3202010

RESEARCH ARTICLE

A Fuzzy Training Framework for Controllable Sequence-to-Sequence Generation

JIAJIA LI¹, PING WANG², ZUCHAO LI^{1,2}, XI LIU³, MASAO UTIYAMA⁴, EIICHIRO SUMITA⁴, HAI ZHAO⁵, AND HAOJUN AI²

¹Music School, Hankou University, Wuhan 430212, China

²Wuhan University, Wuhan 430072, China

³Wuhan Conservatory of Music, Wuhan 430060, China

⁴National Institute of Information and Communications Technology, Kyoto 184-8795, Japan

⁵Shanghai Jiao Tong University, Shanghai 200240, China

Corresponding authors: Ping Wang (wangping@whu.edu.cn) and Zuchao Li (zcli.charlie@gmail.com)

This work was supported by the National Natural Science Foundation of China under Grant 72074171.

ABSTRACT The generation of music lyrics by artificial intelligence (AI) is frequently modeled as a language-targeted sequence-to-sequence generation task. Formally, if we convert the melody into a word sequence, we can consider the lyrics generation task to be a machine translation task. Traditional machine translation tasks involve translating between cross-lingual word sequences, whereas music lyrics generation tasks involve translating between music and natural language word sequences. The theme or key words of the generated lyrics are usually limited to meet the needs of the users when they are generated. This requirement can be thought of as a restricted translation problem. In this paper, we propose a fuzzy training framework that allows a model to simultaneously support both unrestricted and restricted translation by adopting an additional auxiliary training process without constraining the decoding process. This maintains the benefits of restricted translation but greatly reduces the extra time overhead of constrained decoding, thus improving its practicality. The experimental results show that our framework is well suited to the Chinese lyrics generation and restricted machine translation tasks, and that it can also generate language sequence under the condition of given restricted words without training multiple models, thereby achieving the goal of green AI.

INDEX TERMS Music lyrics generation, controllable generation, music understanding, constrained decoding, fuzzy training.

I. INTRODUCTION

Music lyrics combine musical and literary elements. The use of deep learning technologies to generate music lyrics investigates the use of artificial intelligence in artistic creation. Music lyric generation can be defined as a musical melody-conditioned language generation task that spans the two domains of music understanding and language generation. Recently, music understanding has been mainly developed at the audio and symbolic levels. However, since audio is easily affected by many factors, symbols, as an intuitive form of musical description, are more suitable as the basis for music understanding. As a result, the symbolic form of musical melody is used as the input for lyric generation

The associate editor coordinating the review of this manuscript and approving it for publication was Ali Shariq Imran¹.

in this work. By converting musical melody symbols into token sequences, we can model musical lyric generation and machine translation both as a sequence-to-sequence model.

Formally, the music lyrics generation task can be analogized to the machine translation task if the melody input is transformed into the token sequence, as shown in Figure 1. Music melody symbolization is generally represented by MIDI. Similar to natural language text in machine translation, MIDI can be viewed as a sequence of musical events, i.e. tokens of processed natural language text. The main differences lie that a single note can be played for a duration, and multiple notes can be played simultaneously.

In music lyrics generation, specific phrases and words like subject or sentiment are usually proposed to be mentioned in the prediction when the domain of input is known, it is a common and important requirement to improve the

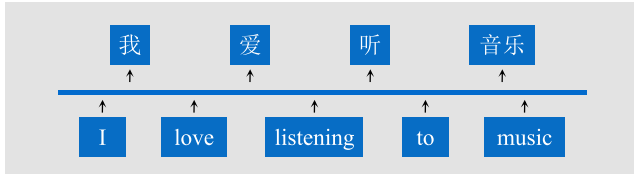
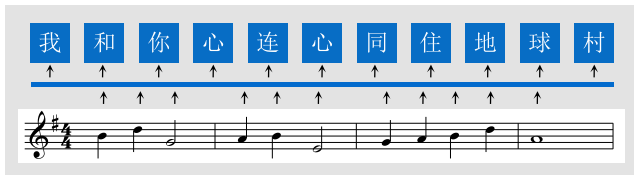
Machine Translation:**Music Lyric Generation:**

FIGURE 1. Analogy figure of machine translation and music lyrics generation. The melody and lyrics in the music lyric generation example comes from the theme song of the 2008 Beijing Olympics “You and Me.”

controllability of the generation. While in machine translation, restricted translation is a special task, whose goal is to generate specific pre-specified terms in the translation output. Therefore, on the basis of treating the music lyrics generation task as machine translation task, we further transform the controllable music lyrics generation task into restricted machine translation task.

Neural machine translation (NMT) has recently entered use because of rapid improvements in its performance [1], [2], [3], [4], [5]. The translation mechanism of an NMT model is a black box because it is a special deep neural network model, which means that translation generation is uncontrollable [6], [7], [8], [9]. Although uncontrollable (or unguaranteed) translation can satisfy basic requirements [10], [11], it is unacceptable in some formal scenarios, particularly for key numbers, time, and proper nouns. To address this concern, the restricted translation task has been proposed [12], [13]. This restricts translation by forcing the inclusion of prespecified words and phrases in the generation output, which enables explicit control over the system output.

Restricted machine translation incorporates human prior knowledge into translation. It restricts the flexibility of the translation to satisfy the demands of translation in specific scenarios. Existing work typically imposes constraints on beam search decoding. Although this can satisfy the requirements overall, it usually requires a larger beam size and far longer decoding time than unrestricted translation, which limits the concurrent processing ability of the translation model in deployment, and thus its practicality. Lexically constrained (or guided) decoding (CD) [13], [14], [15], [16], [17], a modification of beam search, has commonly been used in recent restricted translation studies. Although CD is a reasonable option for restricted translation, its slow decoding limits the practicality of restricted translation. Therefore, we propose a novel training framework for restricted translation that requires only minor changes to the ordinary translation model, to address the inconvenience of the decoding time

overhead caused by additional constraints. In this framework, restricted machine translation is achieved by the model structure instead of the CD.

Specifically, we perform translation in two modes in the training framework: end-to-end translation and restricted translation, and reuse the self-attention and cross-attention in the decoder of the translation model. In the end-to-end translation mode, self-attention adopts incremental attention to the target sequence, and then integrates it with the source representation in the cross-attention. In the restricted translation mode, self-attention simultaneously encodes the target sequence with incremental attention encoding, constrains the word order with non-positional bidirectional attention, and then recursively fuses it with the cross-attention. To make the restricted translation training mode adapt to the training data situation with only parallel sentences available, we propose the Sampled Constraints as Concentration (SCC) training approach. In this approach, we sample the target sequence to simulate the constraint words and impose additional penalties on the loss of these sampled words.

Because the restricted translation is embedded with the model structure and training objective in the translation model trained with our framework, restricted translation is performed without CD. Consequently, the inference speed is substantially increased, which greatly improves the practicality of restricted translation. The effectiveness of our proposed training framework is demonstrated by experiments on both restricted machine translation tasks: (WAT21 En↔Ja) and (WMT14 En→De and En→Fr), and lyric generation task based on our own dataset. Results show that our end-to-end translation model can achieve approximately the same performance as the end-to-end translation baseline; moreover, although it only requires unconstrained decoding, it can achieve performance competitive or even superior with that of the baseline with CD for both machine translation and music lyrics generation tasks.

II. RELATED WORK

A. LYRIC GENERATION

Automated Song Writing (ASW) involves generating melodies and lyrics using machine, with melody-to-lyrics (M2L) generation [18], [19], [20], [21] being one of important tasks. Although trained on rap lyrics, Nguyen *et al.* [22] developed a lyric generator based on natural language generating techniques that can be used to generate various types of songs. Onisawa *et al.* [23] merged a lyric generation component and a music composing component into one system, repeating two functions one after the other until the user's needs are met. Barbieri *et al.* [24] presented a framework of Constrained Markov Processes (CMP) for lyric generation. In comparison to pure Markov models and pure constraint-based techniques, CMP can satisfy both style and structure criteria. Potash *et al.* [25] found that the LSTM model outperformed a Markov baseline model in automatic rap lyric generation tasks, which can generate completely new lyrics but in the same style as a rap singer.

Castro *et al.* [26] introduced a creative lyric generation model by merging two different language training models into one framework which could generate a wide range of unique lyrics that are well-suited to song forms. Saeed *et al.* [27] achieved better performance on poetry and lyrics datasets with GAN framework for creative text generation than generative models based on MLE. Lu *et al.* [28] built a two-channel seq2seq generation model for generating lyrics from a tune, which is more effective since two separate RNNs and big Chinese lyric datasets are employed. Wang *et al.* [29] developed a thematic-aware Seq2Seq model, which is a framework for generating long Chinese lyrics that are well-connected in context. It use LDA to extract the topic so that lyrics may be generated to fit it. For the generation of Chinese song lyrics, Fan *et al.* [30] proposed a novel Seq2Seq model. It performs well in context connection and theme awareness using huge Chinese lyric datasets for training and encoding of multiple levels of contextual information. For more flexible choices to be offered to users, Zhang *et al.* [31] proposed an AI-assisted system for generating high-quality lyrics that allows users to select from a variety of options in generating new lyrics or selecting existing lyrics from context.

Manjavacas *et al.* [32] proposed a method for producing hip-hop lyrics using a neural language model based on RNNs that can be trained end to end. The method used in Nikolov *et al.* [33] for creating rap lyrics has two processes: producing new lyrics from new content and reconstructing rap lyrics from key phrases retrieved from previous text. Vechtomova *et al.* [34] worked on bimodal neural network model with spectrogram VAE and text VAE for generating lyrics from music audio clips. Chen *et al.* [21] proposed an end-to-end system based on SeqGANs to automatically generate lyrics, in which the quality of generated lyrics is not influenced much by a piece of melody or a constrained text theme.

Besides, literature also explored other three tasks for ASW: lyrics-to-lyrics generation (L2L) [35], melody-to-melody generation (M2M) [36], [37], lyric-to-melody generation (L2M) [38], [39].

B. MACHINE TRANSLATION

Vaswani *et al.* [40] proposed a new model that combined neural probabilistic language model [41], rectified linear units, and noise-contrastive estimation, and they integrated it into a machine translation system by reranking k-best lists as well as direct integration into the decoder. Consequently, they improved the scale of NMT model to 1.1B in a large range test across four languages. RNNsearch is a new framework proposed by Bahdanau *et al.* [1], instead of encoding the source sentence into a fixed-length vector, which limits the performance of traditional models, their model automatically searched for parts of a source sentence related to predicted target words. Luong *et al.* [42] proposed two effective frameworks of attention-based NMT: global and local, which outperformed baselines in a variety of translation

tasks. Sennrich *et al.* [43] introduced a method for improving machine translation fluency which employs pseudo parallel training data synthesized from back-translation as an additional parallel dataset while without changes to the normal neural network architecture. A simple but effective method that encodes rare and non-original words as sequences of subword units is proposed by Sennrich *et al.* [44] for improving the machine translation performance of NMT. In comparison to previous back-off dictionary baseline models, subword model achieves better BLEU scores.

Freitag *et al.* [45] presented knowledge distillation as a method for reducing the time cost of student NMT model which obtains better translation quality based on the teacher model. Ahmed *et al.* [46] proposed a weighted Transformer with multiple modified attention layers, which consists of multiple self-attention branches instead of multi-head attention. The resulted model obtained better translation performance compared to the baseline model. Belinkov *et al.* [47] explored two strategies for developing a more robust model that can not only deal with noisy texts but also normal structure of words.

SwitchOut is an approach introduced by Wang *et al.* [48] for augmenting data in NMT training. They randomly substituted parts of words in both the source and target sentences with other random terms from their respective vocabularies, which outperforms previous strategies such as word dropout in experiments on three different translation datasets. According to Ott *et al.* [49], reduced precision and big batch training can yield NMT training speedup. Shen *et al.* [50] developed an evaluation process to find a model that can counter-balance the diversity and quality of generations when compared to various references. It was found that certain types of mixture models are more reliable in contrast when compared to variational models and diverse decoding approaches. Edunov *et al.* [51] revealed that, when compared to BLEU, back-translation improves translation accuracy and produces outputs that are closer to natural text. Nguyen *et al.* [52] developed a data diversification technique with fewer parameters and computations for improving the performance of NMT models by mixing forward and backward model predictions and merging them with original datasets.

C. CONSTRAINED DECODING

Lexically constrained (or guided) decoding (CD), a modification of beam search, has commonly been used in recent restricted translation studies. Specifically, some prespecified words or phrases are forced in translation choice. However, although these approaches can theoretically achieve the goal of restricted translation, existing methods are very expensive in terms of decoding time because of the additional constraints that must be considered in decoding; this limits the practicality of CD. Starting from Post *et al.* [13], in which CD was introduced and utilized in NMT, attempts have been made to reduce the time overhead of CD by the use of dynamic beam allocation. Although the time complexity is formally consistent with that of general beam search,

it remains too inefficient to be used on a large scale [15]. Hu *et al.* [16] further extended CD and improved the throughput of restricted translation systems by using batching in vectorized dynamic beam allocation. Although these efforts have improved the practicality of restricted translation, the decoding speed is still far less than that of ordinary decoding.

III. MUSIC LYRICS GENERATION

The melody of the music is used as input to generate the lyrics. There are two types of music that are currently popular: audio signals and symbolic formats. Although the former can represent various music inputs and easy-to-obtained, since it is too complex and not intuitive, we chose symbolic format as our music melody input.

A. MIDI TO COMPOUND WORD

Although MIDI is a simple and effective format for music melody representation, it is not an optimal solution for music melody understanding. Recently, diverse token representations for MIDI have been proposed, with differences in many aspects such as the factors in MIDI being considered (e.g., melody [53], lead sheet [54], piano [36], [55], or multi-track music [56], [57]) the temporal resolution of the time grid, and the way the advancement in time is notated [36].

REMI [36] is a common representation format that uses [bar] and [position] tokens to place tokens on a metrical grid that uniformly divides a bar into a certain number of positions and assumes symbolic timing. Based on this, Hsiao *et al.* [58] further employed an expansion-compression trick to convert a piece of music to a sequence of compound words by grouping neighboring tokens, significantly shortening the token sequences. As a result, we convert MIDI to the compound word format as our input. According to the practice of Hsiao *et al.* [58], we convert MIDI into 7 types of symbols: *Tempo*, *Chord*, *Bar-beat*, *Type*, *Pitch*, *Duration*, and *Velocity*. For better illustration, we show part of melodies of Joe Hisaishi's "The Sun Also Rises" and transformed MIDI into compound words, as shown in Figure 2. As indicated above, *Tempo* is the speed or pace of a given piece, *Chord* represents any harmonic set of frequencies composed of multiple notes that sound simultaneously, *Bar-beat* is the basic unit of time, and *Type*, *Pitch*, *Duration*, and *Velocity* are perceptual properties of sounds.

B. CONDITIONAL GENERATION AS TRANSLATION

Machine translation takes the source language input $X_{MT} = \{x_1, x_2, \dots, x_n\}$ as input, generates the target language output Y , and the translation model optimization goal is $P(Y|X_{MT})$. The music lyrics generation task is a conditional generation task. It takes the music melody as the input and outputs it as the lyric text sequence. After converting MIDI to compound words, formally, the music melody input is $X_{MS} = \{[x_1^1, x_1^2, \dots, x_1^7], [x_2^1, x_2^2, \dots, x_2^7], \dots, [x_n^1, x_n^2, \dots, x_n^7]\}$ and the generation output is $Y = \{y_1, y_2, \dots, y_m\}$, where $[x_i^1, x_i^2, \dots, x_i^7]$ indicates the i -th compound word. Then the

modeling objective of the music lyrics generation model is $P(Y|X_{MS})$.

The only difference between the two task models is the input, which is a common token in one case and a compound word in the other. As a result, it is only necessary to transform the input embedding layer and concatenate together the embeddings of compound words to form a compound embedding in order to adopt a model consistent with machine translation, i.e. we convert the music lyrics generation as machine translation. Formally, the two objectives unified to $P(Y|E)$, where $E = emb(X_{MT})$ in machine translation and $E = [emb(X_{MS}^1) \oplus emb(X_{MS}^2) \oplus \dots \oplus emb(X_{MS}^7)]$ in music lyrics generation, $emb(\cdot)$ indicates the embedding layer.

C. RESTRICTED GENERATION

Restricted generation is a type of controllable generation in which terms to be included in the generated sequence are pre-specified in order to improve the generated topic relevance or sentiment consistency. Specifically, we define the pre-specified terms as C , the modeling objective for restricted generation is $P(Y|E, C)$. Similarly, this modeling objective is consistent with that of restricted machine translation, so we can adopt a model same with restricted machine translation.

Because data and benchmarks for machine translation are more readily available, and evaluation metrics are more simple and intuitive. Thus, in this work, we first investigate a new model structure and method based on restricted machine translation, and then adapt the method to controllable music lyrics generation to explore the performance of music lyrics generation.

Because, with the exception of minor differences in the embedding layer structure, the music lyrics generation is consistent with machine translation, we will discuss on the model structure and training approach based on machine translation in the following sections.

IV. OUR TRAINING FRAMEWORK

We propose a fuzzy training framework. It refers to models that are exposed to both controllable and uncontrollable language generation settings during training. In our framework, uncontrollable generation is the basis of controllable generation, and constraint words are introduced into language generation as fuzz to improve the generalization ability of the model for both uncontrollable generation and controllable generation. Take NMT as example, our training framework comprises two training subprocesses: end-to-end translation and restricted translation. Recent restricted translation studies have focused mainly on the decoding phase, but we set out to integrate restricted translation into the training phase, which makes the motivation of our work completely different from that of previous studies. Our implementation is based on the existing mainstream Transformer NMT baseline; however, because the training method is independent of the baseline, our training framework can easily be generalized to other NMT models and language generation tasks.

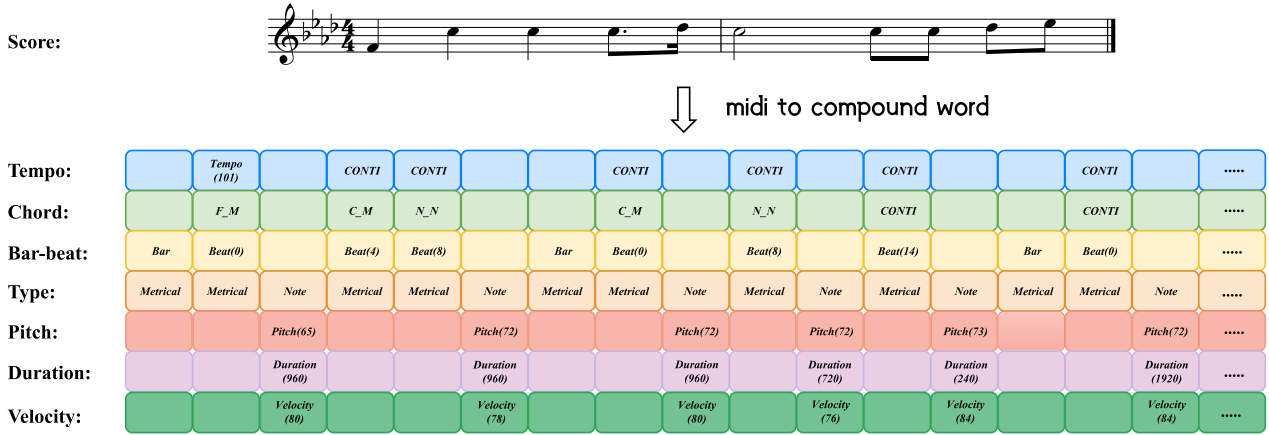


FIGURE 2. Illustration of MIDI to compound word format. The melody comes from "The Sun Also Rises" by Joe Hisaishi.

A. END-TO-END TRANSLATION TRAINING

The most widely adopted form of machine translation is end-to-end translation, which usually employs an encoder-decoder architecture. In the training of end-to-end machine translation, given a source language input $\mathbf{X} = \{x_1, x_2, \dots, x_n\}$ and target language translation $\mathbf{Y} = \{y_1, y_2, \dots, y_m\}$, the model with parameter θ is trained to generate the target output sequence \mathbf{Y} according to the source input sequence \mathbf{X} .

Taking the Transformer model as an example, the encoder is composed of the multi-head self-attention module, whose purpose is to vectorize and contextualize the source input sequence. This module can be formalized as:

$$\mathbf{H}^X = \text{SelfAttn}_{enc}(\mathbf{X} + \text{Pos}(\mathbf{X})), \quad (1)$$

where $\text{Pos}(\cdot)$ represents the position encoding of a sequence, SelfAttn_{enc} denotes the stacked multi-head self-attention encoder, and \mathbf{H}^X is the contextualized source representation. A typical decoder comprises two main components: self-attention and cross-attention. In the self-attention component, the target representation is encoded with similar multi-head attention structures,

$$\hat{\mathbf{H}}^Y = \text{SelfAttn}_{dec}(\text{IncMask}(\hat{\mathbf{Y}} + \text{Pos}(\hat{\mathbf{Y}}))), \quad (2)$$

where $\hat{\mathbf{Y}} = \{BOS, y_1, y_2, \dots, y_{m-1}\}$ is the shifted version of the target sequence \mathbf{Y} , SelfAttn_{dec} denotes the stacked multi-head self-attention layers (similar to the encoder), and IncMask is the extra incremental mask adopted because the sequence on the decoder side is generated incrementally. The target representation is fed to the cross-attention component, as a query, and the source representation is used as the key and value to obtain the final representation, which is then mapped to the target vocabulary space through a linear and softmax layer. The final predicted probabilities can be written as follows:

$$P(\mathbf{Y}) = \text{Softmax}(\text{CrossAttn}(\hat{\mathbf{H}}^Y, \mathbf{H}^X)). \quad (3)$$

The model parameter θ is optimized by minimizing the negative log-likelihood of the gold tokens, according to their predicted probabilities:

$$\begin{aligned} \mathcal{L}_{E2E} &= - \sum_{i=1}^m \log P(y_i) \\ &= - \sum_{i=1}^m \log P(y_i | \mathbf{X}; \hat{\mathbf{Y}}_{<i}; \theta), \end{aligned} \quad (4)$$

where $\hat{\mathbf{Y}}_{<i}$ indicates the sequence before token y_i . In the inference stage, greedy (or beam) search is employed to generate the translation sequence according to predicted probabilities $P(y_i) = P(y_i | \mathbf{X}; \hat{\mathbf{Y}}_{<i}; \theta)$, where $\hat{\mathbf{Y}}$ is the generated token sequence.

B. ILLUSTRATION FOR SAMPLED CONSTRAINTS AS CONCENTRATION

In Figure 3, we show an illustration of the NMT model in the SCC training approach. The illustrated example has input source sentence x_1, x_2, x_3, x_4 ; and target translation y_1, y_2, y_3, y_4 , and tokens previously generated BOS, y_1, y_2, y_3 . In implementation, for efficiency and convenience, we use a padding mask instead of sampling the target translation. Taking y_1 as a sampled constrained word, other tokens are masked, then the original previous output tokens are encoded with incremental attention, and the constrained word sequence is encoded with bidirectional attention. Finally, and the loss for y_1 is additionally penalized to reflect the concentration.

C. TRAINING DETAILS

In our framework, for an input mini-batch, we first use the end-to-end translation training procedure to determine the loss \mathcal{L}_{E2E} . We then sample the target sequence, use the decoder's self-attention component to encode, and reuse the source representation and incremental target representation in the end-to-end translation. It is only necessary to recalculate the cross-attention component to determine the

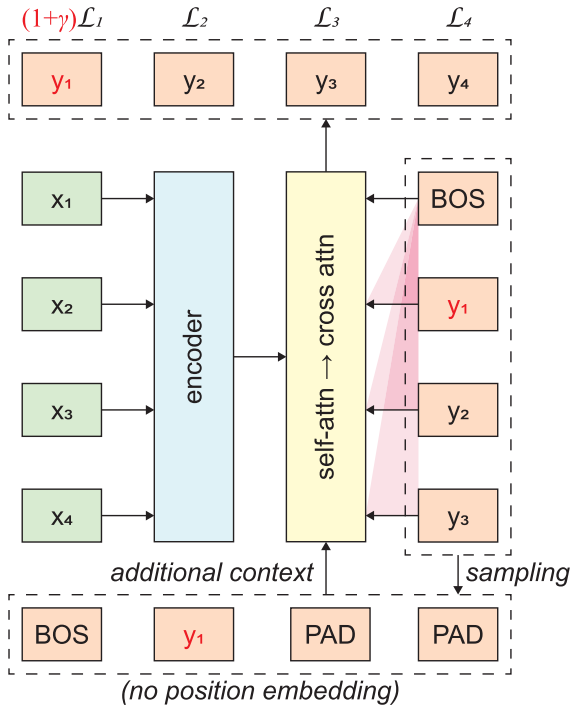


FIGURE 3. The loss for sampled constraints as concentration.

Algorithm 1: Training Procedure

```

1 for  $t = 1, 2, \dots, N$  do
2    $\mathbf{H}^X \leftarrow \text{SelfAttn}_{enc}(\mathbf{X} + \text{Pos}(\mathbf{X}))$ ;
3    $\hat{\mathbf{H}}^Y \leftarrow \text{SelfAttn}_{dec}(\text{IncMask}(\hat{\mathbf{Y}} + \text{Pos}(\hat{\mathbf{Y}})))$ ;
4   End2end Translation
5    $\mathcal{L}_{E2E} \leftarrow \text{Softmax}(\text{CrossAttn}(\hat{\mathbf{H}}^Y, \mathbf{H}^X))$ ;
6   Restricted Translation
7    $\mathbf{C} \leftarrow \text{Sample}(\mathbf{Y})$ ;
8    $\mathbf{H}^C \leftarrow \text{SelfAttn}_{dec}(\mathbf{C})$ ;
9    $\mathcal{L}_{RT} \leftarrow \text{Softmax}(\text{CrossAttn}(\hat{\mathbf{H}}^Y, \mathbf{H}^X) +$ 
    $\text{CrossAttn}(\hat{\mathbf{H}}^Y, \mathbf{H}^C))$ ;
10   $\mathcal{L} \leftarrow \mathcal{L}_{E2E} + \gamma \mathcal{L}_{RT}$ ;

```

loss \mathcal{L}_{RT} . The model parameters are updated in the gradient backward propagation with the two losses added. Because most of the calculations are reused, a small increase in training time makes the trained model suitable for both end-to-end and restricted translation.

We show the training procedure of our proposed framework in Algorithm 1. N is the number of training epochs, and steps 2 and 3 are the reused steps of end-to-end translation and restricted translation training subprocedures. In comparison to ordinary NMT training, the constraint encoding and cross-attention calculation are the only additions our proposed framework requires.

D. RESTRICTED TRANSLATION TRAINING

In recent work on restricted translation, CD, a modification of beam search, has generally been adopted. In CD,

$P(y_i)$ remains unchanged and external search processes are employed, which increases the decoding time overhead. In this paper, we focus on improving the efficiency of restricted translation by modifying $P(y_i)$ to eliminate the additional search processes. Given the constrained word sequence $\mathbf{C} = \{c_1, c_2, \dots, c_k\}$, CD adds additional terms to the predicted probability of the model, and \mathbf{C} is treated as an additional input prompt. The output probability $P(y_i)$ then becomes:

$$P(y_i) = P(y_i | \mathbf{X}; \mathbf{C}; \hat{\mathbf{Y}}_{<i}; \theta). \tag{5}$$

According to this change in the form of probability, we made a simple modification to the workflow of the model, keeping the model structure unchanged. First, we encoded the constrained word sequence with the self-attention component of the decoder. Because the input order of the constrained word sequence is usually inconsistent with the word order of the target sequence, we removed the positional encoding, taking advantage of the position invariance of the self-attention layer. In addition, these constrained words are visible during the entire translation generation process, so there is no need to use the incremental mask strategy. Finally, the constrained words representation is as follows:

$$\mathbf{H}^C = \text{SelfAttn}_{dec}(\mathbf{C}). \tag{6}$$

Regarding such a representation as an additional context, outside of the source representation, the predicted probability of the model can be written as:

$$P(\mathbf{Y}) = \text{Softmax}(\text{CrossAttn}(\hat{\mathbf{H}}^Y, \mathbf{H}^X) + \text{CrossAttn}(\hat{\mathbf{H}}^Y, \mathbf{H}^C)). \tag{7}$$

E. SAMPLED CONSTRAINTS AS CONCENTRATION

The training of end-to-end NMT models generally uses parallel sentences between source and target languages, whereas restricted machine translation requires an additional constraint sequence. To hide the difference between restricted translation training and testing, we propose the SCC training strategy.

Because restricted machine translation training requires additional given constraint sequences, we randomly sample the target sequence to obtain constrained words in this training strategy. However, this is insufficient. Because these additional target words are already exposed to the decoder, the generation of these tokens would become quite easy, and the goal of fully training the model would not be accomplished (i.e., there are shortcuts). This would have an undesirable impact on end-to-end translation (as when no constrained words are prespecified) and reduce the model’s robustness, which is incompatible with our general training framework. Therefore, we propose additional concentration penalties for the losses of these exposed constrained tokens. Denoting the sampled sequence as \mathbf{S}^Y_α , where α is the sampling ratio, and

the penalty factor as γ , the final loss is:

$$\mathcal{L}_{RT} = - \sum_{i=1}^m ((1 + \gamma \mathbb{1}(y_i \in \mathbf{S}_\alpha^Y)) \times \log P(y_i | \mathbf{X}; \mathbf{S}_\alpha^Y; \hat{\mathbf{Y}}_{<i}; \theta)), \quad (8)$$

where $\mathbb{1}(\cdot)$ is the indicator function. Please refer to Appendix IV-B for an illustrated figure and more details.

V. EMPIRICAL EVALUATION AND ANALYSIS

A. SETUP

For basic NMT experiments, our method was evaluated on the ASPEC [59] En↔Ja benchmark¹ and the WMT14 En→De and En→Fr benchmarks. The constrained words for the ASPEC En↔Ja test set were provided by the WAT21 restricted translation shared task and, for WMT14 En→De and En→Fr, we followed previous work by adopting random sampling to extract the constraints. We chose two typical Transformer model settings as our baseline: Transformer-base and Transformer-big, both of which are consistent with Vaswani et al. [3]. During training, we set $\alpha = 0.15$ and $\gamma = 1.0$, the batch size was fixed at 64. For a fair comparison, we utilized beam search strategy in decoding and the decoding beam size was all set to 10.

The framework was implemented using *Fairseq* [60]. In machine translation experiments, the same data preprocessing and subword-splitting [44] script were used for WMT14 En→De and En→Fr as in the *Fairseq* examples. For WAT21 En↔Ja, because of the small size of the ASPEC training set, we also merged the WMT20 En↔Ja and ASPEC training sets to train the model. The joint subword merge size was 44,000 and the other details of preprocessing and the model are consistent with WMT14 En→De. In music lyrics generation, we obtained popular Chinese songs (MP3 format) and lyrics (LRC format) in the past 20 years that are well-known and publicly available on the Chinese Internet, with a total of 2081 pieces of 168-hour audio data. Based on the pre-trained audio to midi transcription model provided by Kong et al. [61], mp3 is transcribed into MIDI format.² Then, we performed MP3-MIDI synchronization, melody extraction, chord recognition, and quantization steps following [58] to obtain normalized compound word format. We perform gap identification for music sequence segmentation according to the time information given in the lyrics. We set the shortest sentence length of the lyrics is 64 and the maximum sentence length is 128. By combining the sentences and their corresponding input that are too short, the input of a single sentence that is over-length will be truncated. Finally, 336 melody-lyric pairs were sampled as the development set, 400 as the test set, and 19,000 as the training set. The preprocess details is shown in Figure 4. We use the character-based

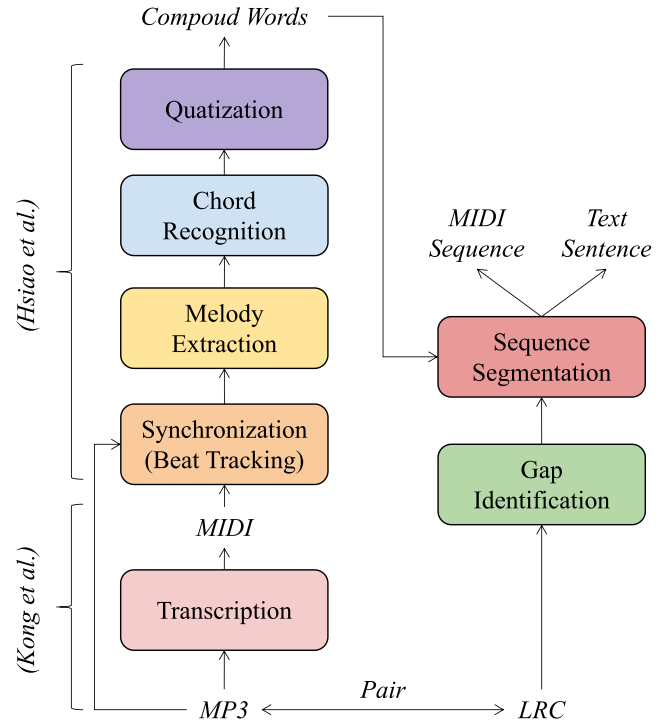


FIGURE 4. Preprocess details of Chinese pop song dataset.

token segmentation to avoid the need for word segmentation in Chinese lyrics. In addition, in the controllable lyrics generation scenario, we sample the nouns that appear in the lyrics based on the part-of-speech to simulate the user’s need for lyrics to contain specific words.

We reported a common BiLingual Evaluation Understudy (BLEU) [62] metric – MultiBLEU scores for automatic evaluation in our machine translation experiments and calculated them using the Moses script. The number of matches is calculated without taking the position into account in the BLEU computation by comparing the n-gram words of the candidate translation with the reference translation. The candidate translation tends to be more accurate the more matches there are. Formally, for each n-gram, the modified precision is calculated as follows:

$$P_n = \frac{\sum_i^E \sum_k^K \min(h_k(c_i), \min_{j \in M} h_k(s_{i,j}))}{\sum_i^E \sum_k^K \min(h_k(c_i))},$$

where s_j indicates the j -th of the M reference translations, c_i indicates the i -th of the E generations, $h_k(c_i)$ represents that the number of n-gram k exists in the generation c_i , and $h_k(s_{i,j})$ represents that the number of n-gram k exists in the reference $s_{i,j}$. The final evaluation metric is obtained by averaging n-gram precision and then multiplied by the length penalty factor.

$$\text{BLEU} = \text{BP} \times \exp\left(\sum_{n=1}^N w_n P_n\right),$$

where **BP** is the brevity penalty to avoid the preference generating short sequences, N is the maximum grams adopted

¹<https://sites.google.com/view/restricted-translation-task/>

²Notably, since Kong et al. [61]’s work is piano-based, we still use their pre-trained model for transcription. Although our dataset is not limited to piano types, we manually checked the MP3-to-MIDI results and found the quality to be acceptable.

TABLE 1. Performance on WAT21 En↔Ja test sets. BS means general beam search, and CD means using constrained decoding. BS+ means beam search with constrained words as an additional context. CD+ stands for constrained decoding with constrained words as an additional context.

| Model | Alg. | En→Ja | | | Ja→En | | |
|-------------|------|--------------|--------------|-----------------|--------------|--------------|-----------------|
| | | BLEU | EM | Speed (sent./s) | BLEU | EM | Speed (sent./s) |
| T-base | E2E | 41.82 | 26.49 | 53.98 | 28.18 | 21.96 | 63.39 |
| | CD | 47.11 | 98.29 | 0.74 | 31.55 | 99.11 | 0.78 |
| Ours | E2E | 41.87 | 26.55 | 53.95 | 28.20 | 22.01 | 63.40 |
| | CAC | 47.15 | 60.26 | 36.01 | 35.46 | 56.68 | 39.32 |
| | CD+ | 47.30 | 98.56 | 0.73 | 35.49 | 99.30 | 0.81 |
| T-big | E2E | 43.33 | 27.51 | 29.68 | 29.45 | 22.70 | 32.53 |
| | CD | 47.89 | 98.30 | 0.68 | 32.04 | 99.16 | 0.71 |
| Ours | E2E | 43.40 | 27.60 | 29.55 | 29.41 | 23.25 | 32.21 |
| | CAC | 47.93 | 60.77 | 18.13 | 35.71 | 57.42 | 19.32 |
| | CD+ | 48.01 | 98.60 | 0.65 | 35.75 | 99.44 | 0.70 |

which is usually set to 4, and $w_n = \frac{1}{N}$ indicates that uniform weights are adopted. For En, De, and Fr, we use the default tokenizer provided by Moses [63], and for Ja, we adopted Mecab³ for word segmentation. In the evaluation of WAT21 EN↔JA, we also reported a consistency metric – the Exact Match (EM) score - according to the WAT21 official instructions. This score is the ratio of sentences in the whole corpus that exactly match the given constraints. For each input, if the characters of the model’s prediction exactly match the characters of (one of) the reference, match = 1, otherwise match = 0.

$$EM = \left(\sum_i^T \max(\text{match}_{i \in E, j \in M}(c_i, s_j)) \right) / T,$$

where T represents the number of examples. For the EM score evaluation, we use lowercase hypotheses and constraints, then use character-level sequence matching (including whitespaces) for each constraint in En, while for Ja, we use character-level sequence matching (including whitespaces) for each constraint without preprocessing. For the evaluation of music lyrics generation, we reported two metrics, Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [64] and Perplexity (PPL). ROUGE is used to evaluate the n-grams overlapping of candidate and reference lyric sentences to judge the quality of the generated lyrics. For language generation tasks, BLEU metric measures the quality of generation based on precision, while ROUGE measures the quality of translations based on recall. Since lyrics generation is a creative generation task, which does not pursue full matching with reference generation compared to machine translation, the ROUGE with longest common subsequence matching (ROUGE-L) metric is thus adopted.

$$R_{LCS} = \frac{LCS(c, s)}{\text{len}(s)}, \quad P_{LCS} = \frac{LCS(c, s)}{\text{len}(c)},$$

$$ROUGE-L = \frac{(1 + \beta^2)R_{LCS}P_{LCS}}{R_{LCS} + \beta^2P_{LCS}},$$

where β is a hyper-parameter, and usually a large number will be set, so ROUGE-L will pay more attention to recall

³<https://taku910.github.io/mecab/>

than precision. Despite different preferences, the ROUGE and BLEU metrics are used to evaluate the matching degree between the model generation and the reference, and besides the matching, the fluency of the language generated by the model is also an important metric, and we adopted PPL to evaluate the fluency of the generated lyrics. It is calculated as follows:

$$PPL = P(w_1, w_2, \dots, w_N)^{-\frac{1}{N}}$$

$$= \sqrt[N]{\prod_{i=1}^N \frac{1}{P(w_i|w_1, w_2, \dots, w_{i-1})}}.$$

It is worth noting that since lyrics are also a kind of artistic creation, the evaluation is relatively subjective and lacks benchmarks, which is not suitable for the model structure exploration stage. Therefore our main basic experiments are based on machine translation tasks due to its diverse datasets and recognized metrics.

B. NMT RESULTS AND ANALYSIS

We present the performance of the models on the WAT21 En↔Ja restricted translation tasks in Table 1. First, for both model architectures (Transformer-base (T-base) and Transformer-big (T-big)), the end-to-end translation performance (E2E) of our approach’s models is almost the same as our baselines. This demonstrates that our training framework still maintains high end-to-end translation performance, even with restricted translation added, meaning it effectively supports both end-to-end translation and restricted translation simultaneously.

Second, on our end-to-end baselines, CD can also be used to accommodate restricted translation. Its very substantial gain in translation performance suggests that CD is a reasonable option for restricted translation. However, under the same conditions, its decoding speed is much lower than that of ordinary decoding, which prevents it from being deployed at a large scale. In our proposed framework, restricted translation is successfully supported with constraints as context (CAC), without using CD. Like CD methods, our method obtains a similar and substantial performance improvement,

but it does so without sacrificing too much decoding speed, which demonstrates that our proposed method is efficient and effective.

Because CAC employs constrained word sequences as additional context, it only imposes soft constraints on the decoder, whereas CD imposes hard constraints. However, because CAC and CD do not conflict, we combined the two as CD+ to produce better results. Our experimental findings attest to the effectiveness of this practice. Furthermore, CAC significantly outperforms CD in Ja→En. This may be due to the beam size of 10, which is insufficient for longer constrained sequences and limits CD performance (a larger beam size will be better, see Figure 1(a)), but our proposed CAC alleviates this shortcoming obviously. Furthermore, for the EM score, CD adheres to hard constraints that the given constrained word must appear in the translation, whereas CAC leverages soft constraints and instead focuses on the overall translation, resulting in a higher BLEU for CAC and a higher EM for CD. CD+, however, provides higher scores for both these metrics.

As in previous studies on restricted translation, we also investigated the impact of constrained words on restricted translation. The constrained words were sampled from the translation references of popular translation datasets (WMT14 En→De and En→Fr). There are five common sampling strategies: *rand1*, *rand2*, *rand3*, *rand4*, and *phr4*. *randk* means that the translation is sampled without replacement *k* times, and *phrk* means that *k* consecutive words are sampled. For a translation length less than *k*, an empty string is output because no constrained words are given.

Table 2 compares the end-to-end translation performance of our T-big model with that of Vaswani et al. [3]’s model. Although we used the same model size and number of training steps, our model’s performance was inferior on En→De but superior on En→Fr. This is a consequence of the use of a larger beam size and the potential benefits of restricted translation training on end-to-end translation. The results also show that the translation performance improved dramatically even when only one constrained word was used. This shows that our method of using constraints as a soft restriction is very effective, and it also demonstrates that translation can be improved substantially with some prior knowledge of translation. The disparities between *rand1* and *rand4* show that accurate prior knowledge of translation can lead to more accurate translation, as the translation uncertainty has been gradually reduced. Additionally, comparing *rand4* and *phr4* demonstrates that the continuous sampling of four constrained words can result in a greater performance improvement than the discrete sampling of four constrained words. This is because *phr4* generally carries more useful information than *rand4*.

C. LYRIC GENERATION RESULTS AND ANALYSIS

After exploring with machine translation tasks, we conduct experimental explorations based on these models on our

TABLE 2. Performance on WMT14 En→De and En→Fr test sets.

| Model | En→De | En→Fr | Speed (sent./s) |
|-----------------------|-------|-------|-----------------|
| Vaswani et al. [3] | 28.40 | 41.80 | – |
| T-big (Ours) | 28.15 | 43.12 | 39.23 / 34.95 |
| +CAC (<i>rand1</i>) | 29.95 | 44.27 | 31.27 / 29.38 |
| +CAC (<i>rand2</i>) | 31.62 | 45.53 | 30.63 / 28.37 |
| +CAC (<i>rand3</i>) | 33.13 | 47.21 | 29.43 / 27.46 |
| +CAC (<i>rand4</i>) | 34.51 | 48.16 | 28.19 / 26.40 |
| +CAC (<i>phr4</i>) | 36.07 | 48.95 | 28.26 / 26.38 |

TABLE 3. Music lyric generation performance on popular song dataset.

| Model | Alg. | ROUGE-L | PPL | Speed (sent./s) |
|--------|------|---------|-------|-----------------|
| GPT-2 | | 6.92 | 31.89 | 13.26 |
| T-base | E2E | 9.62 | 35.73 | 64.55 |
| | CD | 13.55 | 34.59 | 1.92 |
| Ours | E2E | 9.65 | 35.61 | 64.50 |
| | CAC | 12.91 | 33.14 | 39.82 |
| | CD+ | 14.68 | 32.97 | 1.54 |
| T-big | E2E | 10.44 | 33.97 | 41.08 |
| | CD | 15.22 | 32.96 | 1.31 |
| Ours | E2E | 10.39 | 33.81 | 41.17 |
| | CAC | 13.76 | 32.96 | 28.52 |
| | CD+ | 15.90 | 32.73 | 1.02 |

collected lyrics generation task. The evaluation results on music lyrics generation are shown in Table 3.

Under both T-base and T-big model size settings, CD, CAC and CD+ all achieved better generation performance than their corresponding E2E baselines, which is consistent with the conclusion of machine translation, i.e., more deterministic information provided makes the generation process more efficient and controllable, so the quality is better. Second, the CAC method using soft constraints produces similar effects to CD, but with slightly worse scores. The CD+ method combining CAC and CD achieves the best generation results, indicating that the simultaneous incorporation of soft and hard constraints can exert a greater effect than either alone. Third, in terms of decoding speed, CD and CD+ methods bring the largest speed drop, while CAC brings relatively less. And our framework can support both CAC and CD+ restrict generation approaches.

GPT is an advanced language generation solution that has achieved excellent performance in a variety of natural language generation tasks. By ignoring the music information in the input and relying solely on constraints as prompts, restricted music generation can also be regarded as a special natural language generation task. During training, we obtain constraint-lyric pairs by random sampling on the training set to finetune the GPT-2 model. As shown in Table 3, comparing the results of GPT-2 and Seq2seq models, seq2seq models get better ROUGE-L scores, showing that music melody information is essential for lyrics generation. PPL scores for GPT-2 were better than those for Seq2seq models, which indicates that GPT-2 with more language text generation

TABLE 4. Results of ablation study on WAT21 En→Ja test sets.

| Model | En→Ja | Ja→En | Speed (sent./s) |
|-------------------|-------|-------|-----------------|
| T-base (E2E) | 41.82 | 28.18 | 53.98 / 63.39 |
| Ours (CAC) | 47.15 | 35.46 | 36.01 / 39.32 |
| - SCC | 45.63 | 33.05 | 36.05 / 39.25 |
| - RTT | 19.42 | 10.56 | 36.07 / 39.30 |
| + CPos | 43.36 | 29.55 | 35.91 / 39.04 |
| + IncMask | 43.78 | 29.61 | 35.79 / 38.93 |

pre-training can generate more fluent results. It is worth noting that since the model fitting measures cannot reflect completely the performance of generation models, thus our work has limited referential significance on multimedia language generation (MLG).

VI. ABLATION STUDY

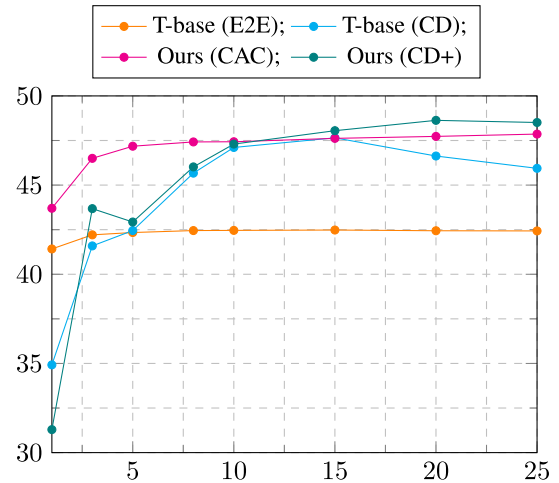
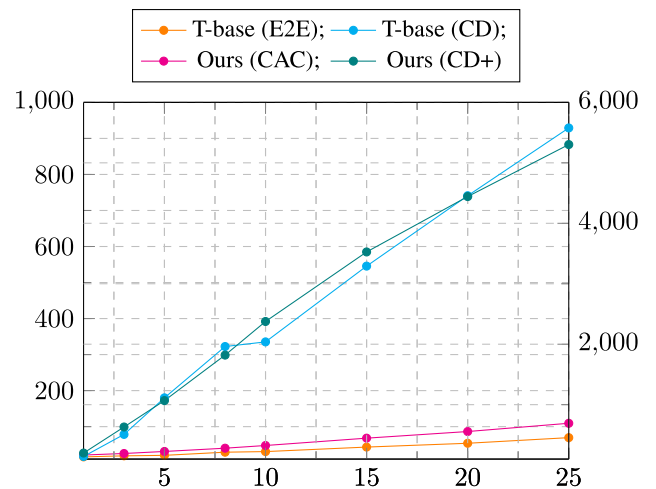
A. INFLUENCE OF BEAM SIZE

To further demonstrate the advantages of our method, we plotted the performance in BLEU score and total decoding time with different beam sizes in Figure 5 and 6.

The results of BLEU score vs. beam size show that, for CD methods or variants (CD+), the translation improves at first as the beam size increases. However, after the beam size increases beyond a certain threshold, the translation performance decreases. Moreover, we have also observed that CD methods require a larger beam size to outperform beam search methods, and they perform worse when beam size is small; because taking additional constraint words into consideration requires more searching. There is no such issue with our CAC method that employs beam search, however.

Figure 6 depicts the total decoding time for various beam sizes. The test set contains 1,812 sentences. We use two y-axes, a larger-scale one on the right to accommodate and denote CD and CD+'s longer decoding times, and a smaller-scale one on the left to denote E2E and CAC's decoding times. The decoding time results show that our CAC method can come close to beam search, a practical restricted translation solution, but CD and CD+ are extremely slow in comparison.

We conducted ablation studies on the model structures and training options of our proposed framework, as shown in Table 4. Using a general MLE loss in restricted translation training; without using SCC loss (-SCC); outperforms the baseline, which shows that the use of restricted translation training can effectively support restricted translation; however, including SCC loss still leads to an improvement over this. This reveals that imposing additional penalties on the loss of constrained words exposed to the decoder is an important design decision. We also evaluated complete removal of the restricted translation training and directly using the end-to-end translation training model for CAC decoding (-RTT). Our results show that the performance greatly suffered, which illustrates the necessity of using restricted translation training for the restricted translation of CAC decoding.


**FIGURE 5.** BLEU score vs. beam size on WAT21 En→Ja test set.**FIGURE 6.** Decoding time vs. beam size on WAT21 En→Ja test set.**TABLE 5.** Influence of melody on music lyric generation performance.

| Model | Setting | ROUGE-L | PPL |
|-------------|--------------|---------|-------|
| Ours | Full | 12.91 | 33.14 |
| | w/o Tempo | 12.53 | 33.07 |
| | w/o Chord | 12.62 | 33.19 |
| | w/o Bar-beat | 12.05 | 32.98 |
| | w/o Type | 12.84 | 33.30 |
| | w/o Pitch | 12.70 | 33.23 |
| | w/o Duration | 12.37 | 32.85 |
| | w/o Velocity | 12.65 | 32.96 |

B. INFLUENCE OF α AND γ

SCC training introduces two new hyper-parameters: the sampling ratio α and the penalty factor γ . To show the impact of these two hyper-parameters on the translation effect, we change one parameter while varying the ($\alpha = 0.15$ and $\gamma = 1.0$ are the static values) and plot the BLEU scores of the resultant models in Figure 7 and 8. We also include the E2E baseline performance (blue line) in the figures for better comparison.

TABLE 6. Case study of music lyrics generation on T-big architecture. The music melody and reference lyrics come from the famous Chinese song “Jasmine Flower.”

| | | | | | | | | | | | | | | | | | | | |
|---------------------|-----------|---|------------|----------------|----------|----------|----------------|----------|------|----------------|----------|-------|----------------|-------|----------|----------------|----------|----------------|-------|
| Scores Input | |  | | | | | | | | | | | | | | | | | |
| Compound Word Input | Tempo: | | Tempo (10) | | CONTI | CONTI | | CONTI | | CONTI | | CONTI | | CONTI | | CONTI | | | |
| | Chord: | | G_M | | G_m | C_m | | CONTI | | N_N | | CONTI | | A_M | | CONTI | | | |
| | Bar-beat: | Bar | Beat(0) | | Beat(4) | Beat(8) | | Beat(12) | | Bar | Beat(0) | | Beat(4) | | Beat(8) | | Beat(12) | | |
| | Type: | Metrical | Metrical | Note | Metrical | Metrical | Note | Metrical | Note | Metrical | Metrical | Note | Metrical | Note | Metrical | Note | Metrical | | |
| | Pitch: | | | Pitch(67) | | | Pitch(67) | | | Pitch(70) | | | Pitch(72) | | | Pitch(75) | | Pitch(75) | |
| | Duration: | | | Duration (960) | | | Duration (480) | | | Duration (480) | | | Duration (480) | | | Duration (480) | | Duration (480) | |
| | Velocity: | | | Velocity (80) | | | Velocity (70) | | | Velocity (70) | | | Velocity (84) | | | Velocity (84) | | Velocity (82) | |
| Reference | | hao yi duo mei li de mo li hua fen fang mei li man zhi ya 好一朵美丽的茉莉花，芬芳美丽满枝丫 (What a beautiful jasmine, with fragrance and beauty all over the branches) | | | | | | | | | | | | | | | | | |
| Baseline [3] | E2E | rang wo men zuo ge mei li de meng xiang shi xia ye de xi yu 让我们做个美丽的梦，像是夏夜的细雨 (Let's have a beautiful dream, like a drizzle on a summer night) | | | | | | | | | | | | | | | | | |
| | CD | wo yuan hua zuo yi duo mo li hua zuo zhe he ni de mei meng 我愿化作一朵茉莉花，做着和你 的美梦 (I would like to turn into a jasmine, have a beautiful dream with you) | | | | | | | | | | | | | | | | | |
| Ours | E2E | qing chun de jiao bu yong bu ting xie zou guo fan hua de da jie 青春 的脚步永不停歇，走过繁华的大街 (The pace of youth never stops, walking through the busy streets) | | | | | | | | | | | | | | | | | |
| | CAC | shei zai qing qing chang zhe mo li de ge san fa chun tian de fen fang 谁在轻轻唱着茉莉的歌，散发春天的芬芳 (Who is gently singing the song of jasmine, sending out the fragrance of spring) | | | | | | | | | | | | | | | | | |
| | CD+ | sheng kai de mo li hua shi chun tian hua de fen fang man ren jian 盛开的茉莉花是春天，花的芬芳满人间 (Jasmine in full bloom is spring, and its fragrance is all over the world) | | | | | | | | | | | | | | | | | |

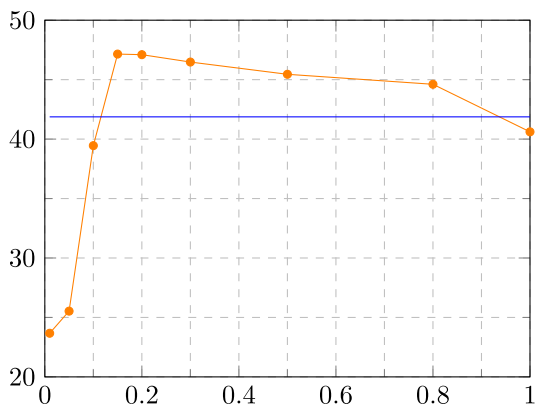


FIGURE 7. BLEU score vs. α on WAT21 En→Ja test set.

Figure 7 shows that a sampling ratio that is too small or too large will prevent RTT from training well. When α is too small, the constrained sequence words used in training and inference differ too much, but when the sample size is too large, the training loss is very small because the majority of the tokens are pre-exposed to the decoder, and the model does not obtain sufficient training. Both situation will corrupt the

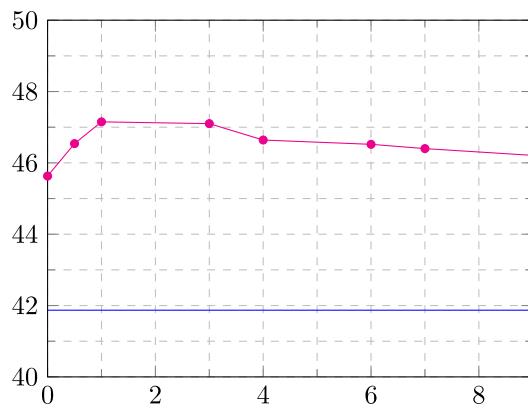


FIGURE 8. BLEU score vs. γ on WAT21 En→Ja test set.

models and are even worse than the baseline. As shown in Figure 8, the penalty factor γ also has an effect on performance. Because it is only used as a penalty, its impact is much lower than that of α . Setting γ to 0.0 effectively removes SCC, but there is also an upperbound on an effective γ , so the best performance (which outperforms the baseline) will be found using a γ that is neither too large nor too small.

C. INFLUENCE OF MELODY FEATURES

The process of generating music lyrics involves two steps: music understanding and language generation. Music melody is converted into a series of compound words forms of seven basic features in compound Transformer, which is then shortened to make it easier to understand. We conducted additional ablation experiments to investigate the influence of these melody features on lyrics generation. As a comparison model, we chose T-base+CAC, and Table 5 list the empirical results. Based on the results in the table, it can be seen that each of the seven features influences the generation effect of lyrics differently. As far as ROUGE-L metric are concerned, *Bar-beat*, *Duration*, and *Tempo* have the greatest impact, while *Type* and *Pitch* have the least. This indicates that lyrics generation is more dependent on rhythmic characteristics. In terms of PPL metric, none of the seven effects are evident, indicating that a single feature does not have significant impact on lyrics generation fluency.

VII. CASE STUDY

Although we utilized the ROUGE-L and PPL metrics to evaluate the quality of lyrics generation, since these metrics cannot intuitively reflect the quality of lyrics generation, we sampled a instance from the test set to performed case study and compared the predicted output of each model with the reference lyric. The results is shown in Table 6.

From the comparison on generations, the reference lyrics are rhyming. In the generated results of baseline, neither the E2E method nor the CD method rhymes (“梦” vs “雨”, “花” vs “梦”). While in our proposed model, except CAC, both E2E and CD+ produce rhyming outputs (“歇” vs “街”, “天” vs “间”), indicating that our framework is more likely to produce rhyme-compliant outputs than baselines. Second, in the CD, CAC and CD+ methods, our pre-specified word is “茉莉花”, where CD and CD+ incorporate explicit constraints, so the word “茉莉花” is included in the generation. The CAC method incorporate an implicit constraint, and the model predict “茉莉” instead of the complete form “茉莉花”, which means that the soft constraint of CAC cannot strictly guarantee that the output contains constraint words, but it comprehensively considers the fluency and the information of the constrained word. Therefore, on the whole, our framework can generate more melodic and rhyming lyrics in the end-to-end mode, and can also trade off the constrained information in the controllable generation mode.

VIII. CONCLUSION AND FUTURE WORK

In this paper, we model music lyrics generation as a machine translation-like task and investigate the problem of controllable generation based on this foundation. Specifically, we proposed novel training and decoding methods for restricted translation and controllable music lyrics generation that do not use CD. Furthermore, we established a general training framework. With our framework, end-to-end generation and constrained generation can be implemented in the same model. Compared to using CD in the end-to-end generation model, we achieved better generation results

including translation and lyrics generation, as well as smaller beam size and consistently higher decoding speed. We evaluated the framework on multiple benchmarks, and demonstrated the performance advantages of constrained generation for controllable music lyrics generation. Using our training framework and decoding method, constrained generation can overcome the limitation of its extremely slow decoding speed and become practical. For our future work, the proposed CAC approach can be further enhanced by applying an attention-over-attention strategy to gain stronger constraint reinforcement and the fuzzy training framework can be used for more controlled language generation tasks. Moreover, since the current music lyric generation dataset suffers from stylistic inconsistencies and lacks word-melody alignment, the lyrics generated by the model must be manually aligned. Our next step will be to label word melody alignment manually in the dataset to certify end-to-end melody lyric filling.

ACKNOWLEDGMENT

An earlier version of this paper was presented in part at the Student Research Workshop of ACL22 [DOI: 10.18653/v1/2022.acl-srw.18].

REFERENCES

- [1] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” in *Proc. ICLR*, 2015, pp. 1–15.
- [2] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, “Convolutional sequence to sequence learning,” in *Proc. ICML*, 2017, pp. 1243–1252.
- [3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
- [4] Z. Li, H. Zhao, R. Wang, M. Utiyama, and E. Sumita, “Reference language based unsupervised neural machine translation,” in *Proc. EMNLP-Findings*, 2020, pp. 4151–4162.
- [5] Z. Li, M. Utiyama, E. Sumita, and H. Zhao, “Unsupervised neural machine translation with universal grammar,” in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2021, pp. 3249–3264.
- [6] A. Moryossef, R. Aharoni, and Y. Goldberg, “Filling gender & number gaps in neural machine translation with black-box context injection,” in *Proc. 1st Workshop Gender Bias Natural Lang. Process.*, 2019, pp. 49–54.
- [7] S. Mehta, B. Azarnoush, B. Chen, A. Saluja, V. Misra, B. Bihani, and R. Kumar, “Simplify-then-translate: Automatic preprocessing for black-box translation,” in *Proc. AAAI*, 2020, pp. 8488–8495.
- [8] Z. Li, R. Wang, K. Chen, M. Utiyama, E. Sumita, Z. Zhang, and H. Zhao, “Explicit sentence compression for neural machine translation,” in *Proc. AAAI*, 2020, pp. 8311–8318.
- [9] R. Miyata and A. Fujita, “Understanding pre-editing for black-box neural machine translation,” in *Proc. EACL*, 2021, pp. 1539–1550.
- [10] Z. Zhang, K. Chen, R. Wang, M. Utiyama, E. Sumita, Z. Li, and H. Zhao, “Neural machine translation with universal visual representation,” in *Proc. ICLR*, 2020, pp. 1–14.
- [11] Z. Li, R. Wang, K. Chen, M. Utiyama, E. Sumita, Z. Zhang, and H. Zhao, “Data-dependent Gaussian prior objective for language generation,” in *Proc. ICLR*, 2020, pp. 1–18.
- [12] C. Hokamp and Q. Liu, “Lexically constrained decoding for sequence generation using grid beam search,” in *Proc. ACL*, 2017, pp. 1535–1546.
- [13] M. Post and D. Vilar, “Fast lexically constrained decoding with dynamic beam allocation for neural machine translation,” in *Proc. NAACL*, 2018, pp. 1314–1324.
- [14] Z. Li, J. Cai, S. He, and H. Zhao, “Seq2seq dependency parsing,” in *Proc. 27th Int. Conf. Comput. Linguistic*, 2018, pp. 3203–3214.
- [15] J. E. Hu, R. Rudinger, M. Post, and B. V. Durme, “PARABANK: Monolingual bitext generation and sentential paraphrasing via lexically-constrained neural machine translation,” in *Proc. AAAI*, 2019, pp. 6521–6528.
- [16] J. E. Hu, H. Khayrallah, R. Culkin, P. Xia, T. Chen, M. Post, and B. Van Durme, “Improved lexically constrained decoding for translation and monolingual rewriting,” in *Proc. NAACL*, 2019, pp. 839–850.

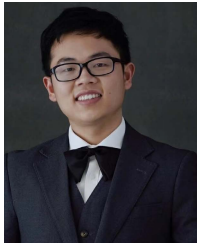
- [17] C. Chen, M. Ye, M. Qi, J. Wu, J. Jiang, and C.-W. Lin, "Structure-aware positional transformer for visible-infrared person re-identification," *IEEE Trans. Image Process.*, vol. 31, pp. 2352–2364, 2022.
- [18] H.-P. Lee, J.-S. Fang, and W.-Y. Ma, "iComposer: An automatic song-writing system for Chinese popular music," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2019, pp. 84–88.
- [19] Z. Sheng, K. Song, X. Tan, Y. Ren, W. Ye, S. Zhang, and T. Qin, "Song-mass: Automatic song writing with pre-training and alignment constraint," in *Proc. AAAI*, 2021, pp. 13798–13805.
- [20] K. Watanabe, Y. Matsubayashi, S. Fukayama, M. Goto, K. Inui, and T. Nakano, "A melody-conditioned lyrics language model," in *Proc. NAACL-HLT*, 2018, pp. 163–172.
- [21] Y. Chen and A. Lerch, "Melody-conditioned lyrics generation with SeqGANs," in *Proc. ISM*, Dec. 2020, pp. 189–196.
- [22] H. Nguyen and B. Sa, "Rap lyric generator," Stanford Univ., Stanford, CA, USA, Tech. Rep., 2009. [Online]. Available: <https://nlp.stanford.edu/courses/cs224n/2009/fp5.pdf>
- [23] C. Nakamura and T. Onisawa, "Music/lyrics composition system considering user's image and music genre," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, Oct. 2009, pp. 1764–1769.
- [24] G. Barbieri, F. Pachet, P. Roy, and M. D. Esposti, "Markov constraints for generating lyrics with style," in *Proc. ECAI*, 2012, pp. 115–120.
- [25] P. Potash, A. Romanov, and A. Rumshisky, "GhostWriter: Using an LSTM for automatic rap lyric generation," in *Proc. EMNLP*, 2015, pp. 1919–1924.
- [26] P. S. Castro and M. Attarian, "Combining learned lyrical structures and vocabulary for improved lyric generation," *CoRR*, vol. abs/1811.04651, 2018. [Online]. Available: <http://arxiv.org/abs/1811.04651>
- [27] A. Saeed, S. Ilić, and E. Zangerle, "Creative GANs for generating poems, lyrics, and metaphors," 2019, *arXiv:1909.09534*.
- [28] X. Lu, J. Wang, B. Zhuang, S. Wang, and J. Xiao, "A syllable-structured, contextually-based conditionally generation of Chinese lyrics," in *Proc. PRICAI*, 2019, pp. 257–265.
- [29] J. Wang and X. Zhao, "Theme-aware generation model for Chinese lyrics," 2019, *arXiv:1906.02134*.
- [30] H. Fan, J. Wang, B. Zhuang, S. Wang, and J. Xiao, "A hierarchical attention based seq2seq model for Chinese lyrics generation," in *Proc. PRICAI*, 2019, pp. 279–288.
- [31] R. Zhang, X. Mao, L. Li, L. Jiang, L. Chen, Z. Hu, Y. Xi, C. Fan, and M. Huang, "Youling: An AI-assisted lyrics creation system," in *Proc. Conf. Empirical Methods Natural Lang. Process., Syst. Demonstrations*, 2020, pp. 85–91.
- [32] E. Manjavacas, M. Kestemont, and F. Karsdorp, "Generation of hip-hop lyrics with hierarchical modeling and conditional templates," in *Proc. 12th Int. Conf. Natural Lang. Gener.*, 2019, pp. 301–310.
- [33] N. I. Nikolov, E. Malmi, C. G. Northcutt, and L. Parisi, "Rapformer: Conditional rap lyrics generation with denoising autoencoders," in *Proc. INLG*, 2020, pp. 360–373.
- [34] O. Vechtomova, G. Sahu, and D. Kumar, "Generation of lyrics lines conditioned on music audio clips," in *Proc. 1st Workshop NLP Music Audio (NLPMusA)*, 2020, pp. 33–37.
- [35] P. Li, H. Zhang, X. Liu, and S. Shi, "Rigid formats controlled text generation," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 742–751.
- [36] Y.-S. Huang and Y.-H. Yang, "Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions," in *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 1180–1188.
- [37] C. A. Huang, A. Vaswani, J. Uszkoreit, I. Simon, C. Hawthorne, N. Shazeer, A. M. Dai, M. D. Hoffman, M. Dinculescu, and D. Eck, "Music transformer: Generating music with long-term structure," in *Proc. ICLR*, 2019, pp. 1–15.
- [38] H. Bao, S. Huang, F. Wei, L. Cui, Y. Wu, C. Tan, S. Piao, and M. Zhou, "Neural melody composition from lyrics," in *Proc. Int. Conf. Natural Lang. Process. Chin. Comput.*, J. Tang, M. Kan, D. Zhao, S. Li, and H. Zan, Eds. Cham, Switzerland: Springer, 2019, pp. 499–511.
- [39] Y. Yu, A. Srivastava, and S. Canales, "Conditional LSTM-GAN for melody generation from lyrics," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 17, no. 1, pp. 1–20, 2021.
- [40] A. Vaswani, Y. Zhao, V. Fossom, and D. Chiang, "Decoding with large-scale neural language models improves translation," in *Proc. EMNLP*, 2013, pp. 1387–1392.
- [41] Y. Bengio, R. Ducharme, and P. Vincent, "A neural probabilistic language model," in *Proc. NIPS*, 2000, pp. 932–938.
- [42] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 1412–1421.
- [43] R. Sennrich, B. Haddow, and A. Birch, "Improving neural machine translation models with monolingual data," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, 2016, pp. 86–96.
- [44] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, 2016, pp. 1715–1725.
- [45] M. Freitag, Y. Al-Onaizan, and B. Sankaran, "Ensemble distillation for neural machine translation," 2017, *arXiv:1702.01802*.
- [46] K. Ahmed, N. S. Keskar, and R. Socher, "Weighted transformer network for machine translation," 2017, *arXiv:1711.02132*.
- [47] Y. Belinkov and Y. Bisk, "Synthetic and natural noise both break neural machine translation," in *Proc. ICLR*, 2018, pp. 1–13.
- [48] X. Wang, H. Pham, Z. Dai, and G. Neubig, "SwitchOut: An efficient data augmentation algorithm for neural machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 856–861.
- [49] M. Ott, S. Edunov, D. Grangier, and M. Auli, "Scaling neural machine translation," in *Proc. 3rd Conf. Mach. Transl., Res. Papers*, 2018, pp. 1–9.
- [50] T. Shen, M. Ott, M. Auli, and M. Ranzato, "Mixture models for diverse machine translation: Tricks of the trade," in *Proc. ICML*, 2019, pp. 5719–5728.
- [51] S. Edunov, M. Ott, M. Ranzato, and M. Auli, "On the evaluation of machine translation systems trained with back-translation," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 2836–2846.
- [52] X.-P. Nguyen, S. Joty, K. Wu, and A. T. Aw, "Data diversification: A simple strategy for neural machine translation," in *Proc. NIPS*, vol. 33, 2020, pp. 10018–10029.
- [53] E. Waite, "Generating long-term structure in songs and stories," *Web Blog Post. Magenta*, vol. 15, no. 4, 2016. [Online]. Available: <https://magenta.tensorflow.org/2016/07/15/lookback-rnn-attention-rnn>
- [54] S. Wu and Y. Yang, "The jazz transformer on the front line: Exploring the shortcomings of AI-composed music through quantitative measures," in *Proc. ISMIR*, 2020, pp. 142–149.
- [55] S. Oore, I. Simon, S. Dieleman, D. Eck, and K. Simonyan, "This time with feeling: Learning expressive musical performance," *Neural Comput. Appl.*, vol. 32, no. 4, pp. 955–967, Feb. 2020.
- [56] C. Payne, "MuseNet," *OpenAI Blog*, vol. 3, Apr. 2019. [Online]. Available: <https://openai.com/blog/musenet/>
- [57] C. Donahue, H. H. Mao, Y. E. Li, G. W. Cottrell, and J. J. McAuley, "LakhNES: Improving multi-instrumental music generation with cross-domain pre-training," in *Proc. ISMIR*, 2019, pp. 685–692.
- [58] W. Hsiao, J. Liu, Y. Yeh, and Y. Yang, "Compound word transformer: Learning to compose full-song music over dynamic directed hypergraphs," in *Proc. AAAI*, 2021, pp. 178–186.
- [59] T. Nakazawa, M. Yaguchi, K. Uchimoto, M. Utiyama, E. Sumita, S. Kurohashi, and H. Isahara, "ASPEC: Asian scientific paper excerpt corpus," in *Proc. LREC*, 2016, pp. 2204–2208.
- [60] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, and M. Auli, "Fairseq: A fast, extensible toolkit for sequence modeling," in *Proc. NAACL-Demo*, 2019, pp. 48–53.
- [61] Q. Kong, B. Li, J. Chen, and Y. Wang, "GiantMIDI-piano: A large-scale MIDI dataset for classical piano music," 2020, *arXiv:2010.07061*.
- [62] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: A method for automatic evaluation of machine translation," in *Proc. 40th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2001, pp. 311–318.
- [63] H. Hoang and P. Koehn, "Design of the Moses decoder for statistical machine translation," in *Proc. Softw. Eng., Test., Quality Assurance Natural Lang. Process. (SETQA-NLP)*, 2008, pp. 58–65.
- [64] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Proc. Workshop Text Summarization Branches Out*, 2004, pp. 74–81.
- [65] Z. Li, M. Utiyama, E. Sumita, and H. Zhao, "Restricted or not: A general training framework for neural machine translation," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics*, 2022, pp. 245–251.



JIAJIA LI received the M.Phil. degree in western music history from the Wuhan Conservatory of Music, Wuhan, China, in 2017. She is currently a Research Assistant with the Music School, Hankou University, after she joined the university, in 2020. Her research interests include traditional Chinese music, intercommunion between Chinese and western music, music composition, and music generation with artificial intelligence.



PING WANG received the B.E. degree in computer science from the Wuhan University of Science and Technology, China, in 2004, the M.M. degree in management from Central China Normal University, China, in 2008, and the Ph.D. degree in information resource management from Wuhan University, China, in 2012. He is currently a Professor with the School of Information Management, Wuhan University. His research interests include data mining and natural language processing.



ZUCHAO LI received the B.S. degree from Wuhan University, Wuhan, China, and the Ph.D. degree from the Center for Brain-Like Computing and Machine Intelligence, Shanghai Jiao Tong University, Shanghai, China, in 2017 and 2022, respectively. He has been an Associate Researcher at the School of Computer Science, Wuhan University, since 2022. He was a Research Fellow at NICT, Japan, from 2019 to 2022. His research interests include natural language processing and related machine learning, data mining, and artificial intelligence.



XI LIU received the B.S. degree from the Wuhan Conservatory of Music, in 2019. Since 2020, he has been a Teaching Assistant with the Wuhan Conservatory of Music. His research interests include Chinese traditional music, music performance, and music generation with artificial intelligence.



MASAO UTIYAMA received the Ph.D. degree from the University of Tsukuba, in 1997. He is currently an Executive Researcher with the National Institute of Information and Communications Technology, Japan. His research interest includes machine translation.



EIICHIRO SUMITA received the B.S. and M.S. degrees in computer science from The University of Electro-Communications, Japan, in 1980 and 1982, respectively, and the Ph.D. degree in engineering from Kyoto University, Japan, in 1999. He has been the Director of the Multilingual Translation Laboratory, National Institute of Information and Communication Technology, since 2006. He worked at the Advanced Telecommunications Research Institute International, from 1992 to 2009, and IBM Research, Tokyo, from 1980 to 1991. His research interests include machine translation and e-learning.



HAI ZHAO received the B.Eng. degree in sensor and instrument engineering and the M.Phil. degree in control theory and engineering from Yanshan University, in 1999 and 2000, respectively, and the Ph.D. degree in computer science from Shanghai Jiao Tong University, China, in 2005. He is currently a Full Professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, after he joined the university, in 2009. He was a Research Fellow at the City University of Hong Kong, from 2006 to 2009, a Visiting Scholar at Microsoft Research Asia, in 2011, and a Visiting Expert at NICT, Japan, in 2012. He is an ACM Professional Member, and served as the Area Co-Chair for ACL 2017 on Tagging, Chunking, Syntax and Parsing, and the Senior Area Chair for ACL 2018 on Phonology, Morphology and Word Segmentation. His research interests include natural language processing and related machine learning, data mining, and artificial intelligence.



HAOJUN AI received the B.S. degree in electronic engineering from Harbin Engineering University, China, in 1994, the M.S. degree in pattern recognition and intelligent control from the Huazhong University of Science and Technology, China, in 1997, and the Ph.D. degree in photogrammetry and remote sensing from Wuhan University, China, in 2003. He is currently an Associate Professor with the School of Cyber Science and Engineering, Wuhan University. His research interests include ubiquitous computing, signal processing, and machine learning.

...