

# Rigid Formats Controlled Text Generation

Piji Li Haisong Zhang Xiaojiang Liu Shuming Shi

Tencent AI Lab, Shenzhen, China

{pijili, hansonzhang, kieranliu, shumingshi}@tencent.com

## Abstract

Neural text generation has made tremendous progress in various tasks. One common characteristic of most of the tasks is that the texts are not restricted to some rigid formats when generating. However, we may confront some special text paradigms such as Lyrics (assume the music score is given), Sonnet, SongCi (classical Chinese poetry of the Song dynasty), etc. The typical characteristics of these texts are in three folds: (1) They must comply fully with the rigid predefined formats. (2) They must obey some rhyming schemes. (3) Although they are restricted to some formats, the sentence integrity must be guaranteed. To the best of our knowledge, text generation based on the predefined rigid formats has not been well investigated. Therefore, we propose a simple and elegant framework named **SongNet** to tackle this problem. The backbone of the framework is a Transformer-based auto-regressive language model. Sets of symbols are tailor-designed to improve the modeling performance especially on format, rhyme, and sentence integrity. We improve the attention mechanism to impel the model to capture some future information on the format. A pre-training and fine-tuning framework is designed to further improve the generation quality. Extensive experiments conducted on two collected corpora demonstrate that our proposed framework generates significantly better results in terms of both automatic metrics and the human evaluation.<sup>1</sup>

## 1 Introduction

Recent years have seen the tremendous progress in the area of natural language generation especially benefiting by the neural network models such as Recurrent Neural Networks (RNN) or Convolutional Neural Networks (CNN) based sequence-to-sequence (seq2seq) frameworks (Bahdanau et al.,

Lyrics	 <p>E - del - weiss, E - del - weiss, ev - ry mor - ning you greet me. Small and white, clean and bright, you look hap - py to meet me.</p>
SongCi	驿外断桥边，寂寞开无主。已是黄昏独自愁，更著风和雨。 无意苦争春，一任群芳妒。零落成泥碾作尘，只有香如故。
Sonnet	Let me not to the marriage of true <i>minds</i> Admit impediments, love is not <i>love</i> Which alters when it alteration <i>finds</i> Or bends with the remover to <i>remove</i> .

Figure 1: Examples of text with rigid formats. In lyrics, the syllables of the lyric words must align with the tones of the notation. In SongCi and Sonnet, there are strict rhyming schemes and the rhyming words are labeled in red color and *italic* font.

2014; Gehring et al., 2017), Transformer and its variants (Vaswani et al., 2017; Dai et al., 2019), pre-trained auto-regressive language models such as XLNet (Yang et al., 2019) and GPT2 (Radford et al., 2019), etc. Performance has been improved significantly in lots of tasks such as machine translation (Bahdanau et al., 2014; Vaswani et al., 2017), dialogue systems (Vinyals and Le, 2015; Shang et al., 2015; Li, 2020), text summarization (Rush et al., 2015; Li et al., 2017; See et al., 2017), story telling (Fan et al., 2018; See et al., 2019), poetry writing (Zhang and Lapata, 2014; Lau et al., 2018; Liao et al., 2019), etc.

Generally, most of the above mentioned tasks can be regarded as free text generation, which means that no constraints on the format and structure, say the number of words and rhyming rules. Note that tasks of dialogue generation and story telling are almost in an open-ending generation style as long as the generated content is relevant with the conditional input text. Although there are

<sup>1</sup>Code: <http://github.com/lipiji/SongNet>

formats constraints on the poetry text, the proposed models just treat the formats as kind of latent information and let the model capture this feature implicitly during training (Liao et al., 2019). The model trained on the five-character quatrain corpus cannot generate seven-character verses. Moreover, it is impossible to trigger these models to generate satisfying results according to arbitrary new defined formats.

In practice we will confront some special text paradigms such as Lyrics (assume the music score is given), Sonnet (say Shakespeare’s Sonnets (Shakespeare, 2000)), SongCi (a kind of Ci. Ci is a type of lyric poetry in the tradition of Classical Chinese poetry.<sup>2</sup>, SongCi is the Ci created during Song dynasty), etc., and some examples are illustrated in Figure 1. The typical characteristics of these text can be categorized into three folds: (1) The assembling of text must comply fully with the predefined rigid **formats**. Assume that the music score is composed, then the lyricist must fill the lyric content strictly tally with the schemes lie in the notation. Take partial of song “Edelweiss” as shown in the first row of Figure 1 as example, the syllables of the lyric words must align with the tones of the notation. The second row of Figure 1 depicts the content of a SongCi created based on the CiPai of “Bu Suan Zi”. Given the CiPai, the number of characters and the syntactical structure of the content are also defined (e.g., the number of characters of each clause: 5, 5. 7, 5. 5, 5. 7, 5.). (2) The arrangement of the content must obey the defined **rhyming** schemes. For example, all the final words (words in red color and *italic* font) of the SongCi content in Figure1 are rhyming (the spelling of each word is: “zhu”, “yu”, “du”, and “gu”). The example in the third row of Figure 1 comes from Shakespeare’s “Sonnet 116” (Shakespeare, 2000), the first four sentences. Usually, the rhyming schemes of Shakespeare’s Sonnets is “ABAB CDCD EFEF GG”<sup>3</sup>. In the example, the rhyming words in scheme “ABAB” are “minds”, “love”, “finds”, and “remove”. (3) Even though the format is rigid, the sentence **integrity** must always be guaranteed. Incomplete sentence such as “love is not the” is inappropriate.

To the best of our knowledge, text generation based on the predefined rigid formats constraints has not been well investigated yet. In this work,

we propose a simple and elegant framework named **SongNet** to address this challenging problem. The backbone of the framework is a Transformer-based auto-regressive language model. Considering the three folds characteristics mentioned above, we introduce sets of tailor-designed indicating symbols to improve the modeling performance, especially for the robustness of the format, rhyme, as well as sentence integrity. We improve the attention mechanism to impel the model to capture the future information on the format to further enhance sentence integrity. Inspired by BERT (Devlin et al., 2019) and GPT (Radford et al., 2018, 2019), a pre-training and fine-tuning framework is designed to further improve the generation quality. To verify the performance of our framework, we collect two corpora, SongCi and Sonnet, in Chinese and English respectively. Extensive experiments on the collected datasets demonstrate that our proposed framework can generate satisfying results in terms of both the tailor-designed automatic metrics including format accuracy, rhyming accuracy, sentence integrity, as well as the human evaluation results on relevance, fluency, and style.

In summary, our contributions are as follows:

- We propose to tackle a new challenging task: rigid formats controlled text generation. A pre-training and fine-tuning framework named SongNet is designed to address the problem.
- Sets of symbols are tailor-designed to improve the modeling performance. We improve the attention mechanism to impel the model to capture the future information to further enhance the sentence integrity.
- To verify the performance of our framework SongNet, we collect two corpora, SongCi and Sonnet, in Chinese and English respectively. We design several automatic evaluation metrics and human evaluation metrics to conduct the performance evaluation.
- Extensive experiments conducted on two collected corpora demonstrate that our proposed framework generates significantly better results given arbitrary formats, including the cold-start formats or even the formats newly defined by ourselves.

## 2 Task Definition

The task of rigid formats controlled text generation is defined as follows:

<sup>2</sup>[http://en.wikipedia.org/wiki/Ci\\_\(poetry\)](http://en.wikipedia.org/wiki/Ci_(poetry))

<sup>3</sup>[http://en.wikipedia.org/wiki/Shakespeare%27s\\_sonnets](http://en.wikipedia.org/wiki/Shakespeare%27s_sonnets)

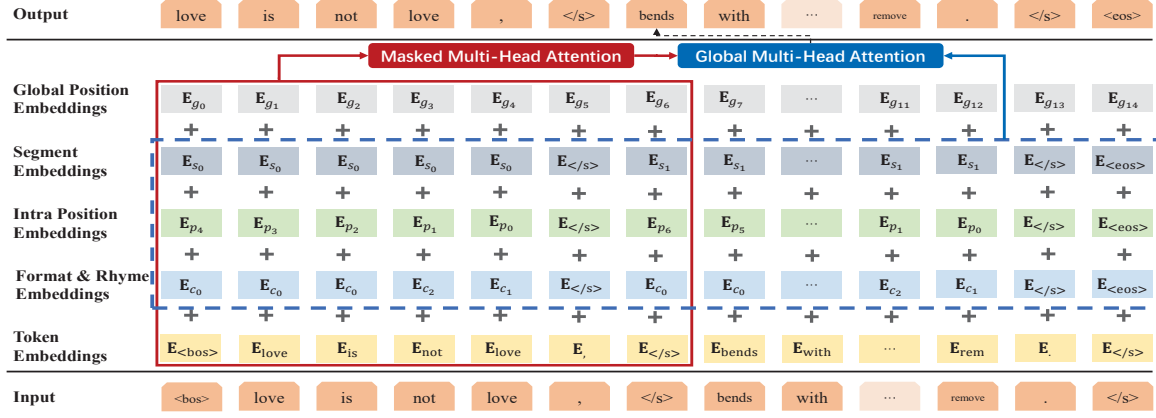


Figure 2: The framework of our proposed model.

**Input:** a rigid format  $C \in \mathcal{C}$ :

$$C = \{c_0 c_1 c_2 c_3, c_0 c_1 c_2 c_3 c_4 c_5\} \quad (1)$$

where  $\mathcal{C}$  is the set of all possible formats. Note that we can define arbitrary new formats not restricted to the ones pre-defined in the corpus, thus  $|\mathcal{C}| \rightarrow \infty$ . Format token  $c_i$  denotes a place-holder symbol of  $C$  which need to be translated into a real word token. Format  $C$  contains 10 words plus two extra punctuation characters “,” and “.”

**Output:** a natural language sentence  $Y \in \mathcal{Y}$  which tally with the defined format  $C$ :

$$Y = \text{love is not love,} \\ \text{bends with the remover to remove.}$$

where the example sentences are extracted from the Shakespeare’s Sonnets (Shakespeare, 2000). From the result  $Y$  we can observe that the count of words is 10 which is consistent with the format  $C$ . The punctuation characters “,” and “.” are also correct. Thus, we claim that it is a 100% format accuracy result. Also, since the two clause sentences are complete, we can get a good sentence integrity score. If  $\mathcal{C}$  is defined on the literary genres of SongCi or Sonnet which have rhyming constraints, the rhyming performance should be evaluated as well. Recall that  $\mathcal{C}$  can be arbitrary and flexible, thus we can rebuild a new format  $C'$  based on the generated result  $Y$  by masking partial content, say  $C' = \{c_0 c_1 c_2 \text{ love, } c_0 c_1 c_2 c_3 c_4 \text{ remove.}\}$ , then we may obtain better results by re-generating based on  $C'$ . We name this operation as **polishing**.

Finally, the target of this problem is to find a mapping function  $G$  to conduct the rigid formats controlled text generation:

$$Y = G(C) \quad (2)$$

### 3 Framework Description

#### 3.1 Overview

As shown in Figure 2, the backbone of our framework is a Transformer-based auto-regressive language model. The input can be the whole token sequences of samples from SongCi or Sonnet. We tailor-design several sets of indicating symbols to enhance the performance in terms of accuracy on format, rhyme, and sentence integrity. Specifically, symbols  $C = \{c_i\}$  are introduced for format and rhyming modeling; Intra-position symbols  $P = \{p_i\}$  are designed to represent the local positions of the tokens within each sentence aiming to improve the rhyming performance and the sentence integrity. Segment symbols  $S = \{s_i\}$  are employed to identify the sentence border to further improve the sentence quality. Attention mechanism is improved to impel the model to capture the future format information such as the sentence ending markers. Similar to BERT (Devlin et al., 2019) and GPT (Radford et al., 2018, 2019), pre-training and fine-tuning paradigm is utilized to boost the performance of the original models.

#### 3.2 Details

We use two sentences (as shown in Figure 1) “love is not love, ..., bends with the remover to remove” extracted from the Shakespeare’s Sonnets (Shakespeare, 2000) as examples to describe the details of our framework SongNet. Since our basic model is a Transformer-based auto-regressive language model, during training, the input is “<bos> love is not love, </s> ..., bends with the remover to remove. </s>”, and the corresponding output is a left-shifting version of the input (tokenized, and we

ignore “...” for convenience and clarity):

*love is not love*,  $\langle /s \rangle$   
*bends with the remover to remove*.  $\langle /s \rangle \langle eos \rangle$

where  $\langle /s \rangle$  denotes the clause or sentence separator, and  $\langle eos \rangle$  is the ending marker of the whole sequence. The target of our framework is to conduct the formats controlled text generation. Therefore, the indicating symbols for format and rhyme as well as the sentence integrity are designed based on the target output sequence.

#### Format and Rhyme Symbols:

$$C = \{c_0, c_0, c_0, c_2, c_1, \langle /s \rangle, c_0, c_0, c_0, c_0, c_0, c_2, c_1, \langle /s \rangle, \langle eos \rangle\} \quad (3)$$

where we use  $\{c_0\}$  to represent the general tokens;  $\{c_1\}$  depict the punctuation characters;  $\{c_2\}$  represent the rhyming tokens “love” and “remove”.  $\langle /s \rangle$  and  $\langle eos \rangle$  are kept.

#### Intra-Position Symbols:

$$P = \{p_4, p_3, p_2, p_1, p_0, \langle /s \rangle, p_6, p_5, p_4, p_3, p_2, p_1, p_0, \langle /s \rangle, \langle eos \rangle\} \quad (4)$$

$\{p_i\}$  denote the local positions of tokens within the same clause or sentence. Note that we align the position symbol indices in a **descending order**. The aim is to improve the sentence integrity by impelling the symbols capture the sentence dynamic information, precisely, the sense to end a sequence. For example,  $\{p_0\}$  usually denote punctuation characters, thus  $\{p_1\}$  should be the ending words of sentences.

#### Segment Symbols:

$$S = \{s_0, s_0, s_0, s_0, s_0, \langle /s \rangle, s_1, s_1, s_1, s_1, s_1, s_1, s_1, \langle /s \rangle, \langle eos \rangle\} \quad (5)$$

where  $s_i$  is the symbol index for sentence  $i$ . The purpose is to enhance the interactions between different sentences in different positions by defining the sentence index features.

During training, all the symbols as well as the input tokens are fed into the transformer-based language model. Contrast to Transformer (Vaswani et al., 2017), BERT (Devlin et al., 2019), and GPT2 (Radford et al., 2019), we modify the traditional attention strategies slightly to fit our problem.

Specifically, for the input, we first obtain the representations by summing all the embeddings of the input tokens and symbols, as shown in the red solid box of Figure 2:

$$\mathbf{H}_t^0 = \mathbf{E}_{w_t} + \mathbf{E}_{c_t} + \mathbf{E}_{p_t} + \mathbf{E}_{s_t} + \mathbf{E}_{g_t} \quad (6)$$

where 0 is the layer index and  $t$  is the state index.  $\mathbf{E}_*$  is the embedding vector for input  $*$ .  $w_t$  is the real token at position  $t$ .  $c$ ,  $p$ , and  $s$  are three pre-defined symbols.  $g$  is the global position index same as position symbols used in Transformer (Vaswani et al., 2017).

Moreover, the state at time  $t$  need to know some future information to grasp the global sequence dynamic information. For example, the model may want to know if it should close the decoding progress by generating the last word and a punctuation character to end the sentence. To represent the global dynamic information, we introduce another variable  $\mathbf{F}^0$  by only summing the pre-defined symbols as shown in the blue dash box of Figure 2:

$$\mathbf{F}_t^0 = \mathbf{E}_{c_t} + \mathbf{E}_{p_t} + \mathbf{E}_{s_t} \quad (7)$$

After processing the input, two blocks of attention mechanisms are introduced to conduct the feature learning procedure. The first block is a masking multi-head self-attention component, and the second block is named global multi-head attention.

#### Masking Multi-Head Self-Attention:

$$\begin{aligned} \mathbf{C}_t^1 &= \text{LN}(\text{FFN}(\mathbf{C}_t^1) + \mathbf{C}_t^1) \\ \mathbf{C}_t^1 &= \text{LN}(\text{SLF-ATT}(\mathbf{Q}_t^0, \mathbf{K}_{\leq t}^0, \mathbf{V}_{\leq t}^0) + \mathbf{H}_t^0) \\ \mathbf{Q}^0 &= \mathbf{H}^0 \mathbf{W}^Q \\ \mathbf{K}^0, \mathbf{V}^0 &= \mathbf{H}^0 \mathbf{W}^K, \mathbf{H}^0 \mathbf{W}^V \end{aligned} \quad (8)$$

where SLF-ATT( $\cdot$ ), LN( $\cdot$ ), and FFN( $\cdot$ ) represent self-attention mechanism, layer normalization, and feed-forward network respectively. Note that we only use the states whose indices  $\leq t$  as the attention context.

After obtaining  $\mathbf{C}_t^1$  from Equation (8), we feed it into the second attention block to capture the global dynamic information from  $\mathbf{F}^0$ .

#### Global Multi-Head Attention:

$$\begin{aligned} \mathbf{H}_t^1 &= \text{LN}(\text{FFN}(\mathbf{H}_t^1) + \mathbf{H}_t^1) \\ \mathbf{H}_t^1 &= \text{LN}(\text{GLOBAL-ATT}(\mathbf{Q}_t^1, \mathbf{K}^1, \mathbf{V}^1) + \mathbf{C}_t^1) \\ \mathbf{Q}^1 &= \mathbf{C}_t^1 \mathbf{W}^Q \\ \mathbf{K}^1, \mathbf{V}^1 &= \mathbf{F}^0 \mathbf{W}^K, \mathbf{F}^0 \mathbf{W}^V \end{aligned} \quad (9)$$

We can observe that all the context information from  $\mathbf{F}^0$  are considered. This is the reason why we name it as “global attention” and why the input real token information  $\mathbf{E}_{w_t}$  is NOT considered. Then

the calculation of the unified first model layer is finished. We can iteratively apply these two attention blocks on the whole  $L$  model layers until obtain the final representations  $\mathbf{H}^L$ . Note that  $\mathbf{H}$  is renewed layerly, however the global variable  $\mathbf{F}^0$  is fixed.

Finally, the training objective is to minimize the negative log-likelihood over the whole sequence:

$$\mathcal{L}^{\text{nl}} = - \sum_{t=1}^n \log P(\mathbf{y}_t | \mathbf{y}_{<t}) \quad (10)$$

### 3.3 Pre-training and Fine-tuning

Although our framework can be trained purely on the training dataset of the target corpus, usually the scale of the corpus is limited. For example, there are only about 150 samples in the corpus of Shakespeare’s Sonnets (Shakespeare, 2000). Therefore, we also design a pre-training and fine-tuning framework to further improve the generation quality.

Recall that in the task definition in Section 2, we claim that our model owns the ability of refining and polishing. To achieve this goal, we adjust the masking strategy used in BERT (Devlin et al., 2019) to our framework according to our definitions. Specifically, we randomly (say 20%) select partial of the original content and keep them not changed when building the format symbols  $C$ . For example, we will get a new symbol set  $C'$  for the example sentences:

$$C' = \{c_0, c_0, c_0, \textit{love}, c_1, \langle /s \rangle, \textit{bends}, c_0, c_0, c_0, c_0, \textit{remove}, c_1, \langle /s \rangle, \langle eos \rangle\}$$

where “love”, “bends” and “remove” are kept in the format  $C'$ .

After the pre-training stage, we can conduct the fine-tuning procedure directly on the target corpus without adjusting any model structure.

### 3.4 Generation

We can assign any format and rhyming symbols  $C$  to control the generation. Given  $C$ , we will obtain  $P$  and  $S$  automatically. And the model can conduct generation starting from the special token  $\langle bos \rangle$  iteratively until meet the ending marker  $\langle eos \rangle$ . Both beam-search algorithm (Koehn, 2004) and truncated top-k sampling (Fan et al., 2018; Radford et al., 2019) method are utilized to conduct the decoding.

## 4 Experimental Setup

### 4.1 Settings

The parameter size of our model are fixed in both the pre-training stage and the fine-tuning stage. The number of layers  $L = 12$ , and hidden size is 768. We employ 12 heads in both the masking multi-head self-attention block and the global attention block. Adam (Kingma and Ba, 2014) optimization method with Noam learning-rate decay strategy and 10,000 warmup steps is employed to conduct the pre-training.

### 4.2 Datasets

We conduct all the experiments on two collected corpus with different literary genres: SongCi and Sonnet, in Chinese and English respectively. The statistic number are shown in Table 3. We can see that Sonnet is in small size since we only utilize the samples from the Shakespeare’s Sonnets (Shakespeare, 2000). Since SongCi and Sonnet are in different languages, thus we conduct the pre-training procedure on two large scale corpus in the corresponding languages respectively. For Chinese, we collect Chinese Wikipedia (1700M Characters) and a merged Chinese News (9200M Characters) corpus from the Internet. We did not conduct the word segmenting operations on the Chinese datasets, which means that we just use the characters to build the vocabulary, and the size is 27681. For English, same as BERT, we employ English Wikipedia (2400M words) and BooksCorpus (980M words) (Zhu et al., 2015) to conduct the pre-training. We did not use BPE operation (Sennrich et al., 2015) on this corpus considering the format controlling purpose. We keep the most frequent 50,000 words to build the vocabulary.

### 4.3 Evaluation Metrics

Besides PPL and Distinct (Li et al., 2016), we also tailor-design several metrics for our task to conduct the evaluation for format, rhyme, and sentence integrity.

**Format** Assume that there are  $m$  sentences defined in the format  $C = \{C_1^s, C_2^s, \dots, C_m^s\}$ , and the generated results  $Y$  contains  $n$  sentences  $Y = \{Y_1^s, Y_2^s, \dots, Y_n^s\}$ . Without loss of generality, we align  $C$  and  $Y$  from the beginning, and calculate the format quality according to the following rules: (1) the length difference  $||C_i^s| - |Y_i^s|| \leq \delta$ ; (2) the punctuation characters must be same. For SongCi, we let  $\delta = 0$  and rule (2) must be conforming.

Model	PPL↓		Diversity (Distinct) ↑			
	VAL	TEST	MA-D-1	MI-D-1	MA-D-2	MI-D-2
S2S	19.61	20.43	75.35	2.48	98.35	36.23
GPT2	148.11	104.99	-	-	-	-
GPT2 w/ Fine-tuning	18.25	17.00	73.87	2.57	96.07	33.92
SongNet (only Pre-training)	24.41	16.23	74.84	4.59	95.09	54.98
SongNet (only Fine-tuning)	12.75	14.73	75.96	2.69	97.59	37.26
SongNet	<b>11.56</b>	<b>12.64</b>	75.04	2.66	97.29	36.78

Model	Format↑		Rhyme↑		Integrity↓
	MA-F1	MI-F1	MA-F1	MI-F1	
S2S	44.32	38.16	53.80	52.27	8.30±2.06
GPT2 w/ Fine-tuning	35.70	35.20	53.48	52.50	45.92±20.12
SongNet (only Pre-training)	29.12	29.46	53.77	53.13	30.98±14.06
SongNet (only Fine-tuning)	99.81	99.83	<b>79.23</b>	<b>78.63</b>	2.14±0.10
SongNet	<b>99.88</b>	<b>99.89</b>	73.21	72.59	<b>1.77±0.16</b>

Table 1: Automatic evaluation results on SongCi

Model	PPL↓		Diversity (Distinct) ↑			
	VAL	TEST	MA-D-1	MI-D-1	MA-D-2	MI-D-2
GPT2 w/ Fine-tuning	31.47	31.03	73.87	2.57	96.07	33.92
SongNet (only Pre-training)	28.56	28.07	49.92	25.14	85.35	65.70
SongNet (only Fine-tuning)	34.62	34.53	42.31	4.96	90.76	47.26
SongNet	<b>27.46</b>	<b>27.63</b>	43.01	10.43	80.06	56.14

Model	Format↑		Rhyme↑		Integrity↓
	MA-F1	MI-F1	MA-F1	MI-F1	
GPT2 w/ Fine-tuning	2.03	1.91	5.20	6.24	15.77±3.63
SongNet (only Pre-training)	99.99	99.99	3.93	4.01	15.28±2.04
SongNet (only Fine-tuning)	99.25	99.99	7.50	7.41	18.86±2.59
SongNet	98.73	98.73	<b>11.46</b>	<b>11.41</b>	<b>11.86±3.01</b>

Table 2: Automatic evaluation results on Sonnet

Corpus	#Train	#Dev	#Test	#Vocab
SongCi	19,244	847	962	5310
Sonnet	100	27	27	2801

Table 3: Statistics of the datasets SongCi and Sonnet.

For Sonnet, we relax the condition where we let  $\delta = 1$  and ignore rule (2). Assume that the number of format-correct sentences is  $n'$ , then we can obtain Precision  $p = n'/n$ , Recall  $r = n'/m$ , and F1-measure. We report both the **Macro-F1** and **Micro-F1** in the results tables.

**Rhyme** For SongCi, usually, there is only one group of rhyming words in one sample. As the example shown in Table 1, the pronunciation of the red rhyming words are “zhu”, “yü”, “du”, and “gu” respectively, and the rhyming phoneme is “u”. For the generated samples, we first use the tool

pinyin<sup>4</sup> to get the pronunciations (PinYin) of the words in the rhyming positions, and then conduct the evaluation. For Shakespeare’s Sonnets corpus, the rhyming rule is clear “*ABAB CDCD EFEF GG*” and there are 7 groups of rhyming tokens. For the generated samples, we employ the CMU Pronouncing Dictionary<sup>5</sup> (Speech@CMU, 1998) to obtain the phonemes of the words in the rhyming positions. For example, the phonemes for word “asleep” and “steep” are [’AH0’, ’S’, ’L’, ’IY1’, ’P’] and [’S’, ’T’, ’IY1’, ’P’] respectively. And then we can conduct the evaluation by counting the overlapping units from both the original words and the extracted phonemes group by group. We report the Macro-F1 and Micro-F1 numbers in the results tables as well. **Integrity** Since the format in our task is strict and

<sup>4</sup><http://github.com/mozillazg/python-pinyin>

<sup>5</sup><http://www.speech.cs.cmu.edu/cgi-bin/cmudict>

Model	PPL↓		Diversity (Distinct) ↑			
	VAL	TEST	MA-D-1	MI-D-1	MA-D-2	MI-D-2
SongNet	<b>12.75</b>	<b>14.73</b>	75.96	2.69	97.59	37.26
SongNet-GRU	16.52	20.49	74.73	1.77	98.30	28.98
SongNet w/o C	13.51	15.38	75.42	2.48	97.36	34.85
SongNet w/o P	14.16	17.16	73.73	2.56	97.52	34.82
SongNet w/ inverse-P	13.40	15.13	74.95	2.54	97.76	35.65
SongNet w/o S	13.23	15.44	75.38	2.74	97.31	37.50

Model	Format↑		Rhyme↑		Integrity↓
	MA-F1	MI-F1	MA-F1	MI-F1	
SongNet	99.81	99.83	79.23	78.63	2.14±0.10
SongNet-GRU	98.99	98.99	52.13	50.93	3.28±1.67
SongNet w/o C	84.73	85.39	78.59	78.24	1.77±0.53
SongNet w/o P	99.61	99.59	67.85	67.29	3.33±0.18
SongNet w/ inverse-P	99.68	99.69	65.89	65.43	2.24±0.21
SongNet w/o S	99.84	99.86	80.43	80.13	1.99±0.10

Table 4: Ablation analysis on SongCi

rigid, thus the number of words to be predicted is also pre-defined. Our model must organize the language using the limited positions, thus sentence integrity may become a serious issue. For example, the integrity of “love is not love .  $\langle /s \rangle$ ” is much better than “love is not the .  $\langle /s \rangle$ ”. To conduct the evaluation of sentence integrity, we design a straightforward method by calculating the prediction probability of the punctuation characters before  $\langle /s \rangle$  given the prefix tokens:

$$Integrity = 2^{-\frac{1}{|Y|} \sum_{i=1}^{|Y|} \log(P(y_{punc}^i | y_0^i, y_1^i, \dots, y_{<punc>}^i))} \quad (11)$$

where  $Y$  is the generated sequence of sentences. Smaller integrity metric value indicates higher sentence quality. To achieve this goal, we conduct pre-trainings for two GPT2 (Radford et al., 2019) models on the large scale Chinese corpus and English corpus respectively. Then we utilize the GPT2 models to conduct the evaluation for sentence integrity.

**Human Evaluations** For SongCi, we sampled 50 samples for 25 CiPais. For Sonnet, the whole 27 samples in the test set are selected for human evaluation. We recruit three helpers to score the *Relevance*, *Fluency*, and *Style*. The rating criteria are as follows: *Relevance*: **+2**: all the sentences are relevant to the same topic; **+1**: partial sentences are relevant; **0**: not relevant at all. *Fluency*: **+2**: fluent; **+1**: readable but with some grammar mistakes; **0**: unreadable. *Style*: **+2**: match with SongCi or

Sonnet genres; **+1**: partially match; **0**: mismatch.

#### 4.4 Comparison Methods

**S2S** Sequence-to-sequence framework with attention mechanism (Bahdanau et al., 2014). We regard the format and rhyme symbols  $C$  as the input sequence, and the target as the output sequence.

**GPT2** We fine-tune the GPT2 models (the pre-training versions are used for sentence integrity evaluation) on SongCi and Sonnet respectively.

**SongNet** Our proposed framework with both the pre-training and fine-tuning stages.

We also conduct ablation analysis to verify the performance of the defined symbols as well as the variants of model structures.

- **SongNet (only pre-tuning)** Without the fine-tuning stage.
- **SongNet (only fine-tuning)** Without the pre-training stage.
- **SongNet-GRU** Employ GRU (Cho et al., 2014) to replace Transformer as the core structure.
- **SongNet w/o C** Remove the format and rhyme symbols  $C$ .
- **SongNet w/o P** Remove the intra-position symbols  $P$ .
- **SongNet w/o S** Remove the sentence segment symbols  $S$ .
- **SongNet w/ inverse-P** Arrange the intra-position indices in ascending order instead of the descending order.

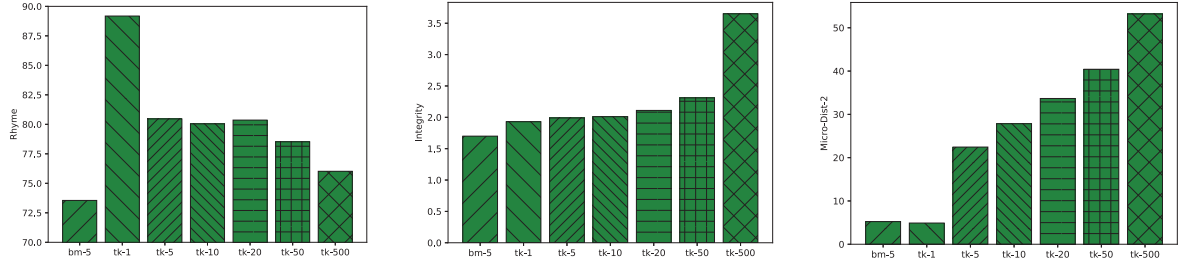


Figure 3: Parameter tuning of  $k$  on the metrics of Rhyme, Integrity, and Micro-Dist-2.

Model	Cases of Generated Results
SongNet-SongCi	<p><b>CiPai:</b> Zhe Gu Tian, <b>Format:</b> 7, 7, 7, 7, 3, 3, 7, 7, 7, 7. 南枝欲破春寒<b>浅</b>(qian)。东风一霎春犹<b>暖</b>(nuan)。海棠零乱暗香<b>残</b>(can)，露颗未消红泪<b>满</b>(man)。春不<b>管</b>(guan)，人何<b>远</b>(yuan)。江头旧日清溪<b>岸</b>(ban)。欲凭楼上数行书，却为梦魂无<b>见</b>(ban)。</p> <p><b>CiPai:</b> Bu Suan Zi, <b>Format:</b> 5, 5, 7, 5, 5, 5, 7, 5. 新月挂帘旌，暗草蛩<b>凄</b>(chu)。夜半银潢十二闾，画幕灯花<b>吐</b>(tu)。何处笛声残，晓窗风<b>雨</b>(yu)。独许寒梅伴小鬟，香在秋千<b>柱</b>(zhu)。</p> <p><b>CiPai:</b> Self-Defined, <b>Format:</b> 3, 3, 5, 3, 3, 5, 7, 7. 翠岩中，流水外，别有小壶<b>天</b>(tian)。自霜清，独斗芳，谁敢并飞<b>仙</b>(xian)。夜久凉生一曲愁，月如花影似花<b>圆</b>(yuan)。</p> <p><b>CiPai:</b> Self-Defined, <b>Format:</b> 9, 9, 9, 9. 雨洒黄泉<b>粼</b>(lin)细作<b>痕</b>(han)。越水西来时节自清<b>闲</b>(xian)。歌笑一杯长唱醉醒<b>间</b>(jian)。春去不堪寻梦绕<b>山</b>(shan)。</p>
SongNet-Sonnet	<p>how do you hold such a thing like this, \ when my eyes are so not black? \ but how can i show myself, so strange, \ that all this black is white? where am i to hide this from my eyes, \ from this white mine eyes all fals, \ where is the good fortune, in me, \ that hath no excuse, no excuse? what is that which can mask the true love \ and for whom is this true love more? \ the one, which shall save the poor my eye, \ from the false truth of my judgment? what lies, for when you are not that, \ no one in this and that can see me lies!</p>

Table 5: Cases of the generated results for SongCi and Sonnet respectively. For SongCi, the number in Format (e.g., 3,5,7) denotes the number of tokens in one sentence. The rhyming words are labeled in red color and *italic* font following is the Pinyin. (Since cases are provided to confirm the format consistency, thus we did not conduct translation for the Chinese samples. Translation for Chinese poetry is also a challenging task.)

Model	Cases of Generated Results Given the Formats with Partial Content
SongNet-SongCi	<p><b>CiPai:</b> Bu Suan Zi, <b>Format:</b> 5, 5, 7, 5, 5, 5, 7, 5.</p> <p><b>Format C:</b> _ _ _ _ _ 到。 _ _ _ _ _ 俏。 _ _ _ _ _ 报。 _ _ _ _ _ 笑。 (1)风暖莺声苦，又是年时<b>到</b>。长爱柳色三分里，多少为春<b>俏</b>。共少年时节，惯得花时<b>报</b>。争似醉径入廛来，步步随春<b>笑</b>。 (2)春来春又去，花影几波<b>到</b>。门外绿杨柳絮洒，溪桥人人<b>俏</b>。门外东风马，似旧时香<b>报</b>。但怪桃李晚时羞，冷落无人<b>笑</b>。</p> <p><b>Format C:</b> _ 雨 _ _ _ _ _ 到。 _ _ _ _ _ 冰， _ _ _ _ _ 俏。 _ _ _ _ _ 春， _ _ _ _ _ 春报。 _ 山花 _ _ _ _ _ 笑。 (1)春雨碧连营，柳暗门巷<b>到</b>。宫殿步辇踏寒<b>冰</b>，百媚生春<b>俏</b>。正日烘晴<b>春</b>，花雾春<b>风报</b>。一点山花醉眼明，半倚阑干<b>笑</b>。 (2)雪雨弄轻暖，新涨绿波<b>到</b>。楼上对面湿翠<b>冰</b>，一夜风花<b>俏</b>。晓日烘长<b>春</b>，睡起春<b>宵报</b>。枕畔山花落叶声，不是梦魂<b>笑</b>。</p>
SongNet-Sonnet	<p>_ _ _ _ _ with _ _ _ hearts , _ _ _ lacking _ _ _ dead ; _ _ _ love _ _ _ parts and _ _ _ buried . _ _ _ many _ _ _ tear , hath _ _ _ eye , _ _ _ now appear, _ _ _ _ _ thee lie ! _ _ _ buried _ _ _ live _ _ _ of _ _ _ gone , _ _ _ parts _ _ _ give _ _ _ thine alone : _ _ _ view _ _ _ thee , _ _ _ _ _ all _ _ _ me .</p> <p>though all thy love <b>with</b> thy <b>hearts</b> , thou still are <b>lacking</b> of my <b>dead</b> ; if thy love <b>love</b> is lost to your love and <b>parts</b> , and yet mine own heart can be <b>buried</b> . so <b>many</b> are ill or in <b>tear</b> , <b>hath</b> not this time that we will make their <b>eye</b> , for that which lies not well hath <b>now appear</b> , no longer nor the world that holds <b>thee lie</b> ! for if it would be <b>buried</b> in my <b>live</b> , or by the earth of mine was <b>gone</b> , then my own <b>parts</b> as my body and mine <b>give</b> , may not be so far beyond <b>thine alone</b> : so far as thee and this world <b>view</b> find <b>thee</b> , then mine life be far enough from <b>all</b> thee and no <b>me</b> .</p>

Table 6: Cases of the generated results given the formats with partial pre-defined content. Format token “\_” needs to be translated to real word token.

## 5 Results and Discussions

### 5.1 Results

Please note that we mainly employ top- $k$  sampling method (Fan et al., 2018; Radford et al., 2019) to conduct the generation, and we let  $k = 32$  here. The parameter tuning of  $k$  is described in Section 5.3.

Table 1 and Table 2 depict the experimental results of SongNet as well as the baseline methods S2S and GPT2 on corpus SongCi and Sonnet respectively. It is obvious that our pre-training and fine-tuning framework SongNet obtain the best per-

formance on most of the automatic metrics. Especially on the metric of Format accuracy, SongNet can even obtain a 98%+ value which means that our framework can conduct the generation rigidly matching with the pre-defined formats. On the metric of PPL, Rhyme accuracy, and sentence integrity, SongNet also performs significantly better in a large gap than the baseline methods such as S2S and GPT2 as well as the model variants only with the pre-training or fine-tuning stage.

Another observation is that some of the results on corpus Sonnet are not as good as the results



Model	Relevance	Fluency	Style
SongNet-SongCi	1.36	1.45	2.00
SongNet-Sonnet	0.58	0.42	0.83

Table 7: Human evaluation results.

on SongCi. The main reason is that Sonnet only contains 100 samples in the training set as shown in Table 3. Therefore, the model cannot capture sufficient useful features especially for the rhyming issue.

## 5.2 Ablation Analysis

We conduct ablation study on corpus SongCi and the experimental results are depicted in Table 4. It should note that all the models are purely trained on SongCi corpus without any pre-training stages. From the results we can conclude that the introduced symbols  $C$ ,  $P$ , and  $S$  indeed play crucial roles in improving the overall performance especially on the metrics of format, rhyme, and sentence integrity. Even though some of the components can not improve the performance simultaneously on all the metrics, the combination of them can obtain the best performance.

## 5.3 Parameter Tuning

Since we employ top- $k$  sampling as our main decoding strategy, thus we design several experiments to conduct the parameter tuning on  $k$ . We let  $k$  to be 1, 5, 10, 20, 50, 500 respectively. We also provide the beam-search (beam=5) results for comparing and reference.

The parameter tuning results are depicted in Figure 3. From the results we can observe that large  $k$  can increase the diversity of the results significantly. But the Rhyme accuracy and the sentence integrity will drop simultaneously. Therefore, in the experiments we let  $k = 32$  to obtain a trade-off between the diversity and the general quality.

## 5.4 Human Evaluation

For human evaluation, we just conduct the judging on the results generated by our final model SongNet. From the result we can observe that the results on corpus SongCi is much better than the ones on corpus Sonnet, which is because the corpus scale is different. And the the small scale also lead to dramatically dropping on all the metrics.

## 5.5 Case Analysis

Table 5 depicts several generated cases for SongCi and Sonnet respectively. For SongCi, the formats (CiPai) are all cold-start samples which are not in the training set or even newly defined. Our model can still generate high quality results on the aspects of **format**, **rhyme** as well as **integrity**. However, for corpus Sonnet, even though the model can generate 14 lines text, the quality is not as good as SongCi due to the insufficient training-set (only 100 samples). *We will address this interesting and challenging few-shot issue in the future.*

In addition, we mentioned that our model has the ability of refining and polishing given the format  $C$  which contains some fixed text information. The examples of the generated results under this setting are shown in Table 6, which show that our model SongNet can generate satisfying results especially on SongCi.

## 6 Conclusion

We propose to tackle a challenging task called rigid formats controlled text generation. A pre-training and fine-tuning framework SongNet is designed to address the problem. Sets of symbols are tailor-designed to improve the modeling performance for format, rhyme, and sentence integrity. Extensive experiments conducted on two collected corpora demonstrate that our framework generates significantly better results in terms of both automatic metrics and human evaluations given arbitrary cold start formats.

## References

- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.
- Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734.
- Zihang Dai, Zhilin Yang, Yiming Yang, William W Cohen, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*.

- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.
- Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898.
- Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1243–1252. JMLR. org.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Philipp Koehn. 2004. Pharaoh: a beam search decoder for phrase-based statistical machine translation models. In *Conference of the Association for Machine Translation in the Americas*, pages 115–124. Springer.
- Jey Han Lau, Trevor Cohn, Timothy Baldwin, Julian Brooke, and Adam Hammond. 2018. Deep-speare: A joint neural model of poetic language, meter and rhyme. *arXiv preprint arXiv:1807.03491*.
- Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119.
- Piji Li. 2020. An empirical investigation of pre-trained transformer language models for open-domain dialogue generation. *arXiv preprint arXiv:2003.04195*.
- Piji Li, Wai Lam, Lidong Bing, and Zihao Wang. 2017. Deep recurrent generative decoder for abstractive text summarization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2091–2100.
- Yi Liao, Yasheng Wang, Qun Liu, and Xin Jiang. 2019. Gpt-based generation for classical chinese poetry. *arXiv preprint arXiv:1907.00151*.
- Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding with unsupervised learning. Technical report, Technical report, OpenAI.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).
- Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389.
- Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083.
- Abigail See, Aneesh Pappu, Rohun Saxena, Akhila Yerukola, and Christopher D Manning. 2019. Do massively pretrained language models make better storytellers? *arXiv preprint arXiv:1909.10705*.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.
- William Shakespeare. 2000. *Shakespeare’s sonnets*. Yale University Press.
- Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1577–1586.
- Speech@CMU. 1998. Carnegie-mellon university pronouncing dictionary for american english. Version 0.7b. Available at [<http://www.speech.cs.cmu.edu/cgi-bin/cmudict>].
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.
- Oriol Vinyals and Quoc Le. 2015. A neural conversational model. *arXiv preprint arXiv:1506.05869*.
- Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.
- Xingxing Zhang and Mirella Lapata. 2014. Chinese poetry generation with recurrent neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 670–680.
- Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.