Training a Chinese RapStar: Applying Rapformer Model to Generate Chinese Rap Lyrics

Stanford CS224N Custom Project

Bihan Liu
Department of Computer Science
Stanford University
bihanliu@stanford.edu

Haoxiang "Mike" Yang
Department of Computer Science
Stanford University
yanghx@stanford.edu

Abstract

Research on lyrics generation has largely focused on unconstrained generation. However, songwriters often start with a specific theme or narrative to expand on. Addressing this, our study emphasizes conditional lyrics generation. We specifically focus on Chinese rap lyrics generation, for two reasons. First, Chinese is intrinsically a challenging and thus fascinating language to work with. Second, rap lyrics has its intriguing rhyming structure and offers a more interesting task for us to tackle. Our model, expanded on Rapformer, a conditional rap generation model for English, achieves a strong balance between thematic coherence and rhyming quality.

1 Key Information to include

- Mentor: Bessie Zhang
- Team Contributions: Mike Yang worked on data processing and training. Bihan Liu worked on inference and evaluation. The team members contributed equally to this project.

2 Introduction

Rap music, originated from New York City in 1970s, has evolved into a global phenomenon that transcends cultural and linguistic barriers. Its storytelling and social commentary nature, coupled with intricate rhyme schemes and wordplay, make rap music a fascinating topic to study in the space of natural language processing. Automated rap lyrics generation, in particular, is a problem that researchers have been exploring. Related papers such as Potash et al. (2015) and Xue et al. (2021) focus on generating the rest of a rap song given the input of the first line. While some of these models achieve impressive rhyming output, their results generally lack the thematic coherence and narrative depth of human-written rap. Nikolov et al. (2020) to our knowledge is the first paper that studies conditional rap generation, producing lyrics based on a short script of movie summaries or news articles. This research inspires us to study conditional rap lyrics generation.

Chinese rap music had little presence in mainstream entertainment, until 2017. The air of a Chinese reality show, Rap of China, attracted millions of hiphop music fans and transformed the rap music landscape. Thousands of talented artists emerged in China, producing rap music in Chinese, one of the most difficult languages to learn. This increased popularity and the challenge of modeling rhymes in Chinese motivate us to study Chinese rap lyrics generation.

In this paper, we combine two areas of interest and study Chinese rap lyrics generation conditional on context. We start with a Hugging Face model that is a pretrained GPT-2 model using a general Chinese corpus. We rely on the conditional generation model, Rapformer, presented in Nikolov et al. (2020) to further train the model using a Chinese rap lyrics data. Rapformer only applies to English. We revise features of this model to capture the nuances of Chinese. We evaluate our model variants

in terms of both how good the generated rhymes are and how relevant the generated outputs are to input content. Our best model achieves high performances in both directions.

Our key contributes include the following. First, to our knowledge, we present the first conditional rap lyrics generation model in Chinese, providing artists a tool that produces both rhyming and thematically coherent rap lyrics. Second, the implementation of Rapformer is not publicly available. We contribute to the NLP and music community by open sourcing our re-implementation of Rapformer and all additional engineering we conduct for Chinese tokenization.

3 Related Work

Prior work on rap lyrics often focus on unconditional generation. Potash et al. (2015) uses LSTM to automatically generate unconstrained rap lyrics. Malmi et al. (2016) bases on RankSVM and neural network to stitch lines from existing rap lyrics. Xue et al. (2021)'s DeepRapper model uses transformers to generate both lyrics and rhythms using a special beat token. While DeepRapper is in Chinese, it is still a unconditional model whose input is only the first line of a song.

Nikolov et al. (2020)'s Rapformer is the closest model for our task. Given a content input, Rapformer extracts the content keywords and generates rap style lyrics that are content relevant. However, it is only trained on English lyrics and the tokenization steps during preprocessing and training are not suitable for Chinese. Its source code is also not publicly available. We draw inspiration from Rapformer, revise the model to address the challenge of rhyme modeling in Chinese, and develop a conditional lyrics generation model for Chinese rap songs.

4 Approach

Overview.[1] Our approach bases on Rapformer's architecture and we modify it to work with Chinese. To achieve conditional generation, we take a four-line rap verse and extract content keywords.[2] Similar to Rapformer, we use four variants for the content extraction step: (1) Original: extract all non-stop words in the order they appear. (2) Shuffle: change the order of the extracted words. (3) Drop: remove 20% of the extracted words. (4) Synonym: replace 20% of the words with synonyms. These variants are intended to enhance the model robustness. We then train a model to learn to reconstruct the original verse conditional on the content words. At inference, we apply a similar extraction step to obtain content words in two test datasets - one is still rap lyrics and the other movie reviews. We generate rap style lyrics based on these words to produce a thematically focused verse. After generation, we apply a rhyme enhancement step.

Line Segmentation. Unlike English, Chinese sentences do not contain spaces, making it much hard to segment and tokenize. For example, "自然语言处理" is a six-character phrase that should actually be segmented into only three chucks, "自然" (Natural), "语言" (Language), "处理" (Processing). Each of the six standalone characters does not make semantic sense in the context. Thus, we experiment with two segmentation methods: one using Google's BERT Tokenizer for Chinese that segments individual characters,[3] and two using `Jieba`,[4] a python library that segments a Chinese sentence into semantically sensical sub-phrases. We run the entire training and inference pipeline and confirm non-sensical output from BERT Tokenizer. Hence in this paper, we only report results using `Jieba`.

Keyword Extraction. After segmenting a lyrical line, we remove stop words using the Python library `stopwordsiso`, which contains a list of Chinese stop words.[5] The remaining segments serve as the input content words. For the Synonym model variant, we rely on the NLTK WordNet CMN to identify and replace synonyms.[6]

Training. Rapformer is trained on a 6-layer Transformer model for its English task. The source code of Rapformer is not publicly available. Hence, we must re-implement the architecture ourselves.

[1] See Rapformer's architecture in Figure 4

[2] A verse is a block of lyrics that rappers typically optimize the rhymes for.

[3] https://huggingface.co/docs/transformers/model$_doc/bert\#transformers.BertTokenizer$

[4] https://github.com/fxsjy/jieba

[5] https://pypi.org/project/stopwordsiso/

[6] https://www.nltk.org/howto/wordnet.html

Rapformer trains on 60,000 rap songs (20x our dataset).[7] Due to computational constraints, we instead start with a pretrained, GPT-2-based, general-purpose Chinese generation model on Hugging Face, `uer/gpt2-chinese-cluecorpussmall`. We use the default model parameters: 12 layers, 768 dimensions, 12 attention heads, 1024 feedforward network dimensions, 0.1 dropouts, and GELU activation. We train the model with input of the following format (in Chinese): "Use these keywords to generate lyrics of one rap verse: {content words} [SEP] {original verse}," where [SEP] is a special separation token. For our training, we use a batch size of 16, a learning rate of 2e-5, and a warmup step of 500. We train for 3 epochs and run validation at the end of each epoch.

Inference. After training, we use each model variant for inference on two test datasets. One dataset is part of the rap training data we split out before training, and another is a movie review dataset (See 5.1). We extract content keywords from each and generate rap style lyrics.

Rhyme Enhancement. We further apply a rhyme enhancement step post-inference, which relies on masked language modeling. This approach is introduced by Rapformer, but we must revise the methodology to adapt for the Chinese language. A unique feature of Chinese characters is that their pronunciation cannot be inferred from their writing. For rhyming, we use the python library `pypinyin` to convert generated lyrics to Pinyin, the standard phoneme of Chinese.[8] For each of the four lines in a generated verse, we replace the last segmented word (using `Jieba`) with a special [MASK] token. We use `google-bert/bert-base-chinese`, a Chinese fill-mask model on Hugging Face, to predict top 10 words at the [MASK] location that would fit into the context. Then, for the first two lines of the verse, we implement a search algorithm that iterates through all 10 x 10 pairs of potential replacements and identify the pair that produces the longest rhyme. Through this enhancement step, we are able to place two longer-rhyming words at the end of the first two lines. For example, the fill-mask model predicts "思考" (Pinyin: Si Kao) and "考虑" (Pinyin: Kao Lv) for the end of the first line, both of which mean "thinking". It also predicts "祈祷" (Pinyin: Qi Dao) and "祈福" (Pinyin: Qi Fu) for the end of the second line, both of which mean "praying". Then the longest-rhyming pair would be "思考" (Pinyin: Si Kao) and "祈祷" (Pinyin: Qi Dao), with the rhyme length of two. Additionally, we make sure the pair does not consist of exactly the same words to eliminate trivial choices. We apply the same method to the second two lines of the verse.

Baselines. We use three models as our baselines for the task of Chinese rap lyrics generation. The first is `uer/gpt2-chinese-lyric`, a popular pretrained Chinese lyrics generator on Hugging Face.[9] Although this model is neither conditional nor focused on rap, the lyrical structure and styling of its Chinese text output still provide us a relevant benchmark to evaluate against. Our second baseline is DeepRapper (Xue et al., 2021). It is also not conditional, but it combines the focuses of Chinese and rap, hence a strong model to compare with in terms of rhyming quality and thematic coherence. Finally, we also use GPT-3.5 Turbo. Having the flexibility of prompting, we are able to test its capability of generating Chinese rap lyrics when given a prompt with the content keywords.

5 Experiments

5.1 Data

For training, we use a GitHub dataset that contains lyrics of 3,000 rap songs from 73 Chinese artists.[10] This dataset is organized at the lyrical line level and has no indicator for verse separation. Thus, we keep only songs whose number of lines is a multiple of four, and assume every chunk of four lines form a rhyming verse. This approach gives us 13,360 verses. Furthermore, some lines contain words of other languages, primarily English. Since our tokenization and rhyming techniques only apply to Chinese, we strip out all English words from the lyrics and remove verses that contain a full line of a English sentence. This preprocessing step allows us to focus on rhyming within Chinese rather than cross-language. We retain 10,471 verses, and split them into training, validation, and test datasets with a 80/10/10 ratio.

For evaluation, in addition to the aforementioned test dataset, we also use Douban Movie Short Comments Dataset, a Kaggle dataset that contains over 2 million comments of 28 movies on a

[7] For reference, our dataset of 3,000 songs takes approximately 32 hours for training and inference.

[8] https://pypi.org/project/pypinyin/

[9] https://huggingface.co/uer/gpt2-chinese-lyric

[10] https://github.com/djwackey/chinese-hiphop-lyrics

Chinese movie review platform.[11] We use this dataset to evaluate how well our model transfers the rap style to non-rap context. The comment length ranges from empty to 200 Chinese characters. To extract quality content words from the comments, we restrict to comments with at least 40 characters. Due to computational constraints, we only sampled 200 eligible comments in our evaluation.

In the later sections, we refer to the rap lyrics test dataset as "Rap Test Data" and movie test dataset as "Movie Test Data".

5.2 Evaluation method

We evaluate our models against the baselines on two aspects: the ability to preserve thematic content and to generate high-quality rhymes. For thematic relevancy, we use BLEU and Overlap. For rhyming quality, we use Rhyme Density and Rhyme Accuracy. These metrics are explained as follows:

- **BLEU:** BLEU evaluates a model's capability of reconstructing original lyrics given the input (Papineni et al., 2002). For conditional models (all our models and GPT-3.5 Turbo), the input is the content keywords. For unconditional models (GPT-2 Chinese Lyrics and DeepRapper), the input in the first line of a verse. To calculate BLEU, we use a simplified unigram:

$$BLEU_1 = \min\left(1, \frac{len(\mathbf{y})}{len(\mathbf{y_0})}\right)\left(\frac{\sum_{1gram\in\mathbf{y}} \min(Count_{\mathbf{y}}(1gram), Count_{\mathbf{y_0}}(1gram))}{\sum_{1gram\in\mathbf{y}} Count_{\mathbf{y}}(1gram)}\right)$$

  where reference $\mathbf{y_0}$ is the original lyrics, and candidates $\mathbf{y}$ are generated outputs.

- **Overlap:** Overlap evaluates a model's capability to preserve content words in the input. This metric only applies to conditional models and is computed as a unigram overlap score presented in Nikolov et al. (2020):

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

  between unique unigrams from the input content words $\mathbf{x}$ and the generated rap verse $\mathbf{y}$.

- **Rhyme Density (RD):** Introduced by Malmi et al. (2016), RD aims to measure the average length of rhymes. For each generated line, RD finds another line within the same verse that has the longest rhyme with the target line and calculates the length of the longest rhyme. The RD of a model is the average of this length across all lines in all verses in the generated outputs.

- **Rhyme Accuracy (RA):** RA measures the ratio of lines that rhyme with an adjacent line on at least the end character.

In addition to the quantitative evaluation, we also engage a professional rapper to qualitatively evaluate a small sample of the generated output. We combine automatic and human evaluations to gain insights into our model performance.

5.3 Experimental details

GPT-2 Chinese Lyrics: For the first baseline, we use the default Hugging Face model configurations and run it on our Rap Test Data without any fine-tuning or parameter modification. The default configurations are: 12 layers, 768 dimensions, 12 attention heads, 1024 feedforward network dimensions, 0.1 dropouts, and GELU activation. This inference process takes approximately 1 hour.

DeepRapper: For the second baseline, we run DeepRapper's source code directly on our Rap Test Data without modification to its model configurations.[12] DeepRapper is based on GPT-2 and uses the same default parameters as GPT-2 Chinese Lyrics. This inference process takes approximately 18 hours.

GPT-3.5 Turbo: For the third baseline, we use OpenAI's paid API. We input the system prompt "You will be provided a list of Chinese words, and your task is to generate four lines of Chinese rap

[11] https://www.kaggle.com/datasets/utmhikari/doubanmovieshortcomments

[12] https://github.com/microsoft/muzic/tree/main/deeprapper

lyrics using the words provided," and feed along the content words from Rap Test Data. We make no changes to the default parameters. Due to the cost of GPT-3.5 Turbo, we only evaluate this model with a sample of 20% of the Rap Test Data. This inference process takes approximately 20 minutes.

Our Models: During training, we train on `uer/gpt2-chinese-cluecorpussmall` use the default configurations: 12 layers, 768 dimensions, 12 attention heads, 1024 feedforward network dimensions, 0.1 dropouts, and GELU activation. For fine-tuning with our rap data, we use a batch size of 16, a learning rate of 2e-5, and a warmup step of 500. We train for 3 epochs and run validation at the end of each epoch. For each of the four variants of our model, the entire training process takes approximately 6 hours, and the inference process takes 2.

5.4 Results

Table 1 presents the evaluation results for Rap Test Data. RE stands for models with Rhyme Enhancement. For thematic relevancy, all of our models perform better than GPT-2 Chinese Lyrics and DeepRapper in terms of BLEU. This makes sense because our models feature conditional generation and these two baselines do not. This result is not a perfect apple-to-apple comparison, but demonstrates our models' ability to generate thematically coherent lyrics based on content words. GPT-3.5 Turbo, our baseline conditional model, performs better than all our models in terms of BLEU and Overlap. But we expect this result due to the powerful capability of GPT-3.5 Turbo. In fact, for BLEU, our best model Original demonstrates close performance to GPT-3.5 Turbo, displaying promising capability.

For rhyming quality with RD and RA, our best model Original RE performs much better than GPT-2 Chinese Lyrics and GPT-3.5 Turbo. DeepRapper is better than all our models. We also expect this result because DeepRapper has a much more complex architecture trained on over 300K songs (100x our training data). Also, the inference step of DeepRapper on our Rap Test Data takes 18 hours, while each of our models only requires 2 hours. Results demonstrate that our lightweight models are capable of generating fairly quality rap lyrics in Chinese compared with other options.

Model	BLEU	Overlap	RD	RA
BASELINE				
GPT-2 Chinese Lyric	0.1256	–	0.8120	0.4712
DeepRapper	0.0885	–	**1.4679**	**0.9636**
GPT-3.5 Turbo	**0.3798**	**0.4578**	0.4100	0.3229
OUR MODELS				
Original	**0.3042**	**0.3092**	0.8746	0.3927
Shuffle	0.2742	0.2744	0.8435	0.3876
Synonym	0.2682	0.2782	0.8020	0.3843
Drop	0.2474	0.2443	0.7319	0.3606
Original RE	0.2932	0.2962	**1.1420**	**0.4951**
Shuffle RE	0.2682	0.2665	1.0725	0.4771
Synonym RE	0.2614	0.2684	1.0354	0.4788
Drop RE	0.2405	0.2347	1.0114	0.4746

Table 1: Rap Reconstruction

Table 2 presents evaluation results on our models when the input data is Movie Test Data. Since we are transferring content from movie reviews to rap lyrics, we do not report the BLEU score between these two fundamentally different content structure. Hence, we also do not report results for our baselines given they are not positioned for the style transfer task.

While the Original model is still the best at Overlap, Drop RE now is the best model for rhyming quality. Compared with Table 1, generating rap lyrics using content words from movie reviews demonstrates similar Overlap, but much lower RD and RA. This result is likely due to the differences in nature of extracted content words. Extracted keywords from the rap verses likely already include words that rhyme, which is not the case for movie reviews. Hence, it is logical that generating rhyming lyrics using content words with already good options produces better results. Regardless, an RD of 0.8642 implies that on average, each line rhymes with another line for 0.8642 characters. This result is still promising as it is not too far from 1.0.

Model	Overlap	RD	RA
Original	**0.3185**	0.5455	0.2858
Shuffle	0.2633	0.5537	0.3063
Synonym	0.2906	0.5316	0.2875
Drop	0.2536	0.5584	0.3088
Original RE	0.3120	0.7652	0.3742
Shuffle RE	0.2601	0.7425	0.3875
Synonym RE	0.2848	0.7702	0.4000
Drop RE	0.2484	**0.8642**	**0.4338**

Table 2: Style Transfer from Movie Reviews

To gain more qualitative insight on our models, we invite a professional rapper for human evaluation. We present to him 10 generated rap verses one at a time. For each verse, we ask him "do you think this verse is written by human or generated by AI?" For 10 verses, the professional rapper find 2 of them difficult to distinguish. In Figure 1, we show a verse that is flagged as AI-generated, and in Figure 2 a verse that is flagged as human-written/uncertain.

Original Verse
飘落的雪花 覆盖滚烫的球场。见证了童年的记忆 被时间陆续收藏。走向漫长的走廊 透过光 迎接授奖。后方亲人的厚望在播放失败休想。
Prompt
用以下关键词写一段说唱歌词：飘落；雪花；覆盖；滚烫；球场；见证；童年；记忆；陆续；收藏；走向；漫长；走廊；透过；授奖；亲人；厚望；失败；休想。
Generated Rap
piao luo zai xue gai gun tang de qiu chang jian zheng guo tong nian dui wo ji yi li zui chang de yi ci
飘落在雪花里滚烫的球场是我见证过的童年对我记忆里最长的一次
Falling into the snowflakes, the scorching playground is the childhood I have witnessed, the longest memory I have
suo yi zhe ye jian zheng le shi jian de lu xu shou cang zou xiang ni kan bian de man chang
所以这也见证了时间的陆续收藏走向你看遍的漫长
It also witnesses the continuous collection of time towards the extensive length you have seen
zou zai man chang de zou lang yi ge you yi ge
走在漫长的走廊一个又一个
Walking along the lengthy corridor, one by one, everyone
hou wang suo you ren dou zai wei wo men bo fang yi chang wei wo men wei ni er de
厚望所有人都在为我们播放一场为我们为你而为的
Everyone is broadcasting a play for us, for you

Figure 1: Sample Rap

Original Verse
没有谁我都照样快乐生活。感谢你曾经送给我的折磨。或许能看清楚这是非的错。但在这里的已不是我。
Prompt
用以下关键词写一段说唱歌词：没有；照样；生活；曾经；送给；折磨；或许；看清楚；是非；错。
Generated Rap
sheng huo mei you ren zhao yang mei you shen me da cuo
生活没有人照样没有什么大错
Life goes on without people, and it's not too wrong
dan ye mei you ren kan qing chu
但也没有人看清楚
But there's also no one to see it clearly
ye xu zeng jing dou bei ren song gei guo zhe mo
也许曾经都被人送给过折磨
Perhaps it's better than having once been given torment by others.
huo xu kan qing chu shi fei dui cuo
或许看清楚是非对错
Perhaps seeing clearly what's right and wrong

Figure 2: Sample Rap with Rhyme Enhancement

6 Analysis

The outputs of our models demonstrate promising capabilities for the task of Chinese rap lyrics generation conditional on input content. Figure 3 provides a visualization of our models' performance compared to the baselines, when using Rap Test Data. The axes are RD and RA, and the size of the circle represents the BLEU score, where for greater BLEU score we use bigger circle size. Based on the results, we build a set of models that maintains a strong balance between preserving thematic content and producing quality rhymes, achieving our task fairly well.

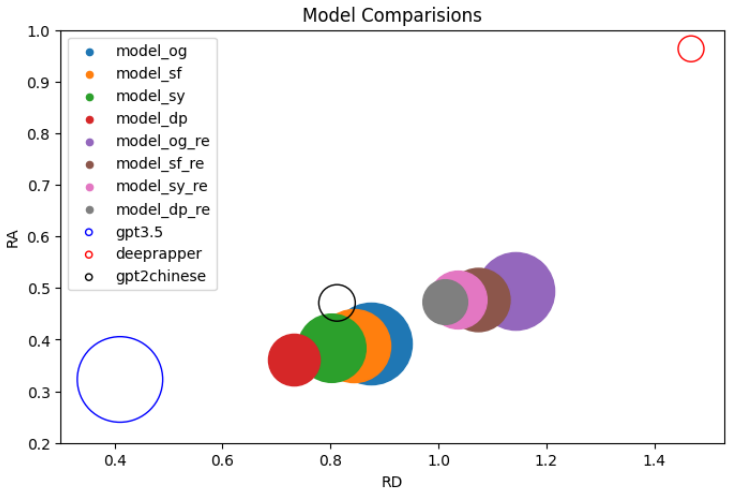

Figure 3: Model Performance Comparisons

We see that for any model variant, the rhyme enhancement step always increases RD and RA and decreases BLEU and Overlap. This result makes sense because this step is supposed to improve rhyming quality, and it does so by replacing some content words with outside choices. Our evaluation demonstrates the effectiveness of the rhyme enhancement.

When using Rap Test Data for rap reconstruction, we find one surprising result that the Original model, regardless the presence of rhyme enhancement, performs better than other model variants across all metrics; the Drop model consistently performs the worst. This result does not align with the original Rapformer paper. On the other hand, when using Movie Test Data for style transfer, we see distinct advantages for different variants. Our hypothesis is that Chinese is inherently a more difficult language to rhyme and rap with. English words are formed by roots. Thus, in English it's easier to find two words that are similar in pronunciation, meaning, and syntactic function at the same time. A simple example is that all verbs can rhyme if in the -ing form. Hence, shuffling, dropping, or replacing some words with synonyms will potentially add robustness to the original English Rapformer. However, this is rarely the case for Chinese. When a deliberately chosen word in the original lyrics is relocated, dropped, or replaced, it is challenging to find a suitable alternative. At the same time, when using Movie Test Data, the content is not rhyming anyways, so different variants may indeed introduce more robustness. Additional analyses could be conducted on a larger scale to test the hypothesis: using more data, a more comprehensive segmentation algorithm for Chinese, or a larger dictionary for synonyms.

7 Conclusion

In this paper, we develop a set of models that generate Chinese rap lyrics conditional on a short corpus of text in any format, e.g., lyrics, movie reviews, or news articles. This task is interesting and relevant because it reflects how human rappers write songs: starting with a theme and pieces of ideas, then developing rhyming lyrics. Compared with other lyrics generation models, our models achieve a strong balance between preserving the theme of the input content and producing high-quality rhymes. Through this process, we tackle several key challenges. First, we research and apply multiple

tools needed to tokenize and process Chinese words, a more difficult task than English. Second, we re-implement Rapformer ourselves given we have no access to the source code and adapt it to generate Chinese content. Meanwhile, our work involves some limitations. First, we train our model using only 3,000 songs. We expect a larger size of the training data would improve performance significantly. Second, we do not conduct thorough qualitative evaluation of the generated lyrics due to time and resource constraints. But in our evaluation, we observe a few generated outputs that have extremely high RD, but in reality just repeating a line multiple times. We remove these rows manually in our results, but do not develop a comprehensive methodology for such issue. Future study should examine and reduce these anomalies from the model systematically.

We highlight our achievement that we beat GPT-3.5 Turbo in terms of rhyming in Chinese. GPT-3.5 Turbo is greatly capable of rhyming in English. But our achievement demonstrates that even the most advanced language model struggles to produce as high-quality results in languages other than English. This conclusion inspires us to further explore LLMs for different languages and motivates us to consider cultural diversity in the space of AI research. Additionally, while we hope to build a tool that helps musicians' creative process, we understand the potential issues in copyrights and originality. We urge future research to explore these ethical challenges and develop aligned AI models.

References

Eric Malmi, Pyry Takala, Hannu Toivonen, Tapani Raiko, and Aristides Gionis. 2016. Dopelearning: A computational approach to rap lyrics generation. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16. ACM.

Nikola I Nikolov, Eric Malmi, Curtis G Northcutt, and Loreto Parisi. 2020. Rapformer: Conditional rap lyrics generation with denoising autoencoders. *arXiv preprint arXiv:2004.03965*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Annual Meeting of the Association for Computational Linguistics*.

Peter Potash, Alexey Romanov, and Anna Rumshisky. 2015. GhostWriter: Using an LSTM for automatic rap lyric generation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1919–1924, Lisbon, Portugal. Association for Computational Linguistics.

Lanqing Xue, Kaitao Song, Duocai Wu, Xu Tan, Nevin L. Zhang, Tao Qin, Wei-Qiang Zhang, and Tie-Yan Liu. 2021. Deeprapper: Neural rap generation with rhyme and rhythm modeling.

A Appendix

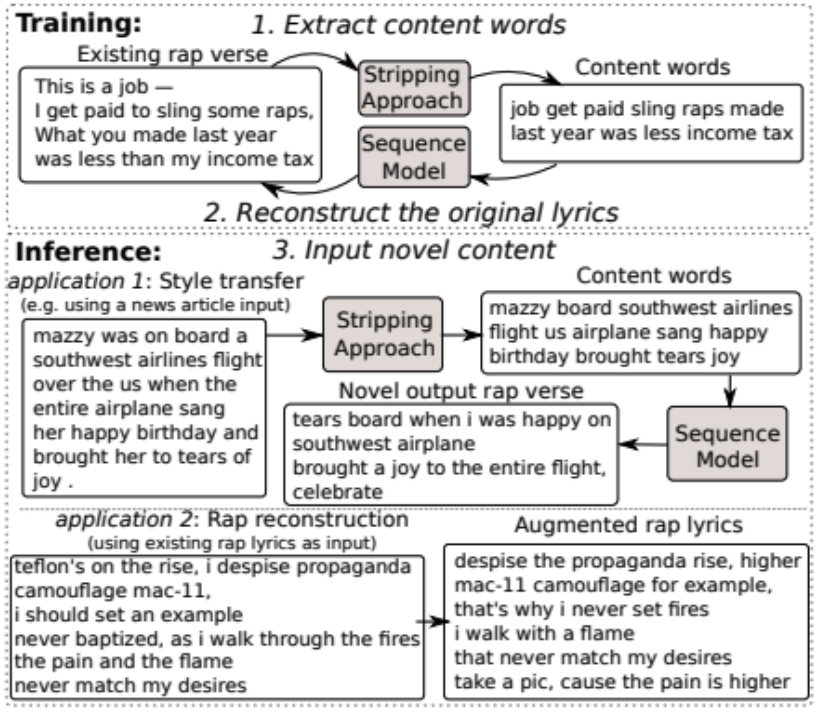

Figure 4: Overview of the approach from Nikolov et al. (2020)

A.1 Different View Metrics

	Rap Reconstruction				Style Transfer from Movies		
Model	**BLEU**	**Overlap**	**RD**	**RA**	**Overlap**	**RD**	**RA**
Model og	**0.3042**	**0.3092**	1.0787	0.3927	**0.3185**	0.6833	0.2858
Model sf	0.2742	0.2744	0.9614	0.3876	0.2633	0.5537	0.3063
Model sy	0.2682	0.2782	1.2073	0.3843	0.2906	0.5863	0.2875
Model dp	0.2474	0.2443	0.9268	0.3606	0.2536	0.6675	0.3088
Model og re	0.2932	0.2962	1.3439	**0.4951**	0.3120	0.8275	0.3742
Model sf re	0.2682	0.2665	1.1873	0.4771	0.2601	0.7425	0.3875
Model sy re	0.2614	0.2684	**1.4308**	0.4788	0.2848	0.8250	0.4000
Model dp re	0.2405	0.2347	1.1943	0.4746	0.2484	**0.9625**	**0.4338**

Table 3: Model mertics for comparison all-together without filtering RD

A.2 Generated Chinese Rap Examples

A.2.1 Baselines

GPT-2 Chinese Lyric:

Frist Line:
你就像冰激凌一样的融化
Generated Rap:
藏在我的梦里生根发芽
有我在黑夜你不会再害怕
请爱我
别这样自私的我还是会爱着你
请爱我
别这样的自私的

Note that GPT-2 Chinese Lyric model baseline are likely to produce duplicated sentences to reach higher rhyme density.

DeepRapper:

Frist Line:
饶舌歌手只是你的街坊邻居
Generated Rap:
他不会讲些华丽的大道理
饶舌押韵发出真实的街头声音
没有光鲜亮丽的外型
在我身上我有了一种超能力
许多美女手牵手却不知道我喜欢你
歌里总是说我是你的唯一

GPT3.5:

Original Rap:
又不见记不得
微笑是苦是甜
能想起跑过住宅区的个单元
把我们赶回家的是羊肉汤香味
没洗手不许吃饭姥姥打手背的竹签
Generated Rap:
在住宅区里我跑得好快
回到家里我想起姥姥的羊肉汤香味
不见的微笑记不得的苦甜
洗手不许吃饭却在手背上留了竹签

Note that most rap verses generated from GPT3.5 are not that good in terms of rhyme quality.

A.2.2 Our Models

Our models on rap reconstruction:

Original Rap:
不用你来担心这都没关系
不用你的关心我做着专辑
不用变的烦心我是个传奇
看我如何单刀匹马打到关底
Generated Raps:
不用每当我担心的关心或者没关系
也不用这么快变得多烦人
我的专辑变得多烦心
让自己别单刀匹马
把我打到一定的关底

Our models on style transfer from movies:

Movie Comments:
不要管那些青红皂白。为了活着自己闷，中国没有的。除了爸妈没人会真心帮你。为了活着，你永远只能够靠你自己。
Generated Raps:
话多但不要吞
没人会真心帮你
你不如永远不能够
除了爸妈没人帮你
就算别人也做不到

Our models with rhyme enhancement on rap reconstruction:

Original Rap:
你是否跟我同样想过
在困难缠绕的每天
虚度年华金钱乱花
不就等于用颓废和堕落消费着时间
Generated Raps:
同样还是想过自己的困难
缠绕又是什么对于自己来说
也是这满是虚幻
也许人会对痛苦不再勇敢
为了每天看着自己的书
也许同样为了金钱的挑战

Our models with rhyme enhancement on style transfer from movies:

Movie Comments:
那些泛黄的照片定格了多少瞬间。最遥远的距离肯定不是你在天边。有没有一个房间给我。暂时忘记理想也忘记现实。
Generated Raps:
泛黄的照片但定格的瞬间是那么的遥远
没有那个距离是否也会这么甘愿
这也许就是那么的不能够为了忘记
也许这就是那个不能够的理想
我的现实走远