Unsupervised Melody-to-Lyric Generation

Yufei Tian[1]*, **Anjali Narayan-Chen**[2], **Shereen Oraby**[2], **Alessandra Cervone**[2], **Gunnar Sigurdsson**[2], **Chenyang Tao**[2], **Wenbo Zhao**[2], **Tagyoung Chung**[2], **Jing Huang**[2], **Nanyun Peng**[1,2]
[1] University of California, Los Angeles, [2] Amazon Alexa AI
{yufeit,violetpeng}@cs.ucla.edu
{naraanja,orabys,cervon,gsig,chenyt}@amazon.com
{wenbzhao,tagyoung,jhuangz}@amazon.com

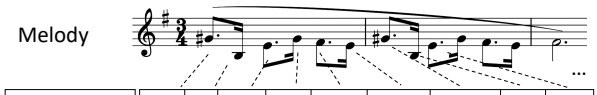
Abstract

Automatic melody-to-lyric generation is a task in which song lyrics are generated to go with a given melody. It is of significant practical interest and more challenging than unconstrained lyric generation as the music imposes additional constraints onto the lyrics. The training data is limited as most songs are copyrighted, resulting in models that underfit the complicated cross-modal relationship between melody and lyrics. In this work, we propose a method for generating high-quality lyrics without training on any aligned melody-lyric data. Specifically, we design a hierarchical lyric generation framework that first generates a song outline and second the complete lyrics. The framework enables disentanglement of training (based purely on text) from inference (melody-guided text generation) to circumvent the shortage of parallel data.

We leverage the segmentation and rhythm alignment between melody and lyrics to compile the given melody into decoding constraints as guidance during inference. The two-step hierarchical design also enables content control via the lyric outline, a much-desired feature for democratizing collaborative song creation. Experimental results show that our model can generate high-quality lyrics that are more on-topic, singable, intelligible, and coherent than strong baselines, for example SongMASS (Sheng et al., 2021), a SOTA model trained on a parallel dataset, with a 24% relative overall quality improvement based on human ratings. [1]

1 Introduction

Music is ubiquitous and an indispensable part of humanity (Edensor, 2020). Self-serve songwriting has thus become an emerging task and has received interest by the AI community (Sheng et al., 2021; Tan and Li, 2021; Zhang et al., 2022; Guo et al., 2022). However, the task of melody-to-lyric (M2L) generation, in which lyrics are generated based on a given melody, is underdeveloped due to two major challenges. First, there is a limited amount of melody-lyric aligned data. The process of collecting and annotating paired data is not only labor-intensive but also requires strong domain expertise and careful consideration of copyrighted source material. In previous work, either a small amount (usually a thousand) of melody-lyrics pairs is manually collected (Watanabe et al., 2018; Lee et al., 2019), or Sheng et al. (2021) use the recently publicized data (Yu et al., 2021) in which the lyrics are pre-tokenized at the syllable level leading to less sensical subwords in the outputs.



Music Note	L	S	L	S	L	S	L	S	L
Human	Ma-	ny	skies	've	turned	to	grey	be-	cause
Baseline 1	Hey,	ba-	by	what	ain't	nothing	wrong to	hide	away
Baseline 2	So-	me	one	got	to	-	go	-	-
Lyra (Ours)	Take	me	back	to	old-	er	time	in	lit-

Figure 1: An example of the melody and the corresponding lyrics, where 'L' denotes a music note with long duration and 'S' stands for short. Our model LYRA generates more coherently than the baselines. Besides, the rhythms of lyrics (i.e., accents and relaxations when spoken) generated by human and LYRA align well with the flows of the melody. On the other hand, existing methods output lyrics that have low singability by either aligning multiple words with one single note (baseline 1) or vice versa (baseline 2) as highlighted in red.

Another challenge lies in melody-to-lyric modeling. Compared to unimodal sequence-to-sequence tasks such as machine translation, the latent correlation between lyrics and melody is difficult to learn. For example, Watanabe et al. (2018); Lee et al. (2019); Chen and Lerch (2020); Sheng et al. (2021) apply RNNs, LSTMs, SeqGANs, or Transformers with melody embeddings and cross attention (Vaswani et al., 2017), hoping to capture the melody-lyrics mapping. However, as shown in Fig-

*Work was done when the author interned at Amazon.
[1]Our code is available at https://github.com/amazon-science/unsupervised-melody-to-lyrics-generation.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 9235–9254

ure 1, these methods may generate less singable lyrics when they violate too often a superficial yet crucial alignment: one word in a lyric tends to match one music note in the melody (Nichols et al., 2009). In addition, their outputs are not fluent enough because they are neural models trained from scratch without leveraging large pre-trained language models (PTLMs).

In this paper, we propose **LYRA**, an unsupervised, hierarchical melody-conditioned LYRics generAtor that can generate high-quality lyrics with content control *without training on melody-lyric data*. To circumvent the shortage of aligned data, LYRA leverages PTLMs and disentangles training (pure text-based lyric generation) from inference (melody-guided lyric generation). This is motivated by the fact that plain text lyrics under open licenses are much more accessible (Tsaptsinos, 2017; Bejan, 2020; Edmonds and Sedoc, 2021), and prior music theories pointed out that the knowledge about music notes can be compiled into constraints to guide lyric generation. Specifically, Dzhambazov et al. (2017) argue that it is the *durations* of music notes, not the pitch values, that plays a significant role in melody-lyric correlation.

As shown in Figure 1, the segmentation of lyrics should match the segmentation of music phrases for breathability. Oliveira et al. (2007); Nichols et al. (2009) also find that long (short) note durations tend to associate with (un)stressed syllables. However, existing lyric generators, even when equipped with state-of-the-art neural architectures and trained on melody-lyrics aligned data, still fail to capture these simple yet fundamental rules. In contrast, we show that through an inference-time decoding algorithm that considers two melody constraints (segment and rhythm) without training on melody-lyrics aligned data, LYRA achieves better singability than the best data-driven baseline. Without losing flexibility, we also introduce a factor to control the strength of the constraints.

In addition, LYRA adopts the hierarchical text generation framework (i.e., plan-and-write (Fan et al., 2019; Yao et al., 2019)) that both helps with the coherence of the generation and improves the controllability of the model to accommodate user-specified topics or keywords. During training, the input-to-plan model learns to generate a plan of lyrics based on the input title and salient words, then the plan-to-lyrics model generates the complete lyrics. To fit in the characteristics of lyrics and melody, we also equip the plan-to-lyrics model with the ability to generate sentences with a predefined count of syllables through multi-task learning.

Our contributions are summarized as follows:

- We design LYRA, the first melody-constrained neural lyrics generator *without training on parallel data*. Specifically, we propose a novel hierarchical framework that disentangles training from inference-time decoding, which is supported by music theories. Our method works with most PTLMs, including those black-box large language models (LLMs) when finetuning is replaced by in-context learning.
- The hierarchical generation design of LYRA enables content or topic control, a feature of practical interest but missing among existing works.
- Both automatic and human evaluations show that our unsupervised model **LYRA** outperforms fully supervised baselines in terms of both text quality and musicality by a significant margin. [2]

2 Background and Problem Setup

Representation of Melody Melody is a succession of pitches in rhythm consisting of a sequence of music phrases, which can be further decomposed into timed music notes. Each music note is defined by two independent pivots: pitch values and durations. *Pitch* represents the highness/lowness of a musical tone; *duration* is the note's length of time. Namely, melody $\mathcal{M}$ can be denoted by $\mathcal{M} = \{p_1, p_2, ...p_M\}$, where each p_i ($i \in 1, 2, ..., M$) is a music phrase. The music phrase can be further decomposed into timed music notes ($p_i = \{n_{i1}, n_{i2}, ...n_{iN_i}\}$), where each music note n_{ij} ($j \in \{1, 2, ..., N_i\}$) comes with a duration and is associated with or without a pitch value. When a music note comes without a pitch value, it is a rest that indicates the absence of a sound and usually aligns with no lyrics.

Task Definition Our goal is to achieve unsupervised melody-to-lyrics generation. We follow the definition of "unsupervised" Machine Translation (MT) tasks (Lample et al.; Artetxe et al., 2019) which achieve cross-lingual translation by training on monolingual data only. In our case, we achieve melody-to-lyrics generation by training on text data only and do not require any parallel melody-lyrics aligned data for training.

[2] Examples of lyrics generated by the complete pipeline can be found in this demo page.

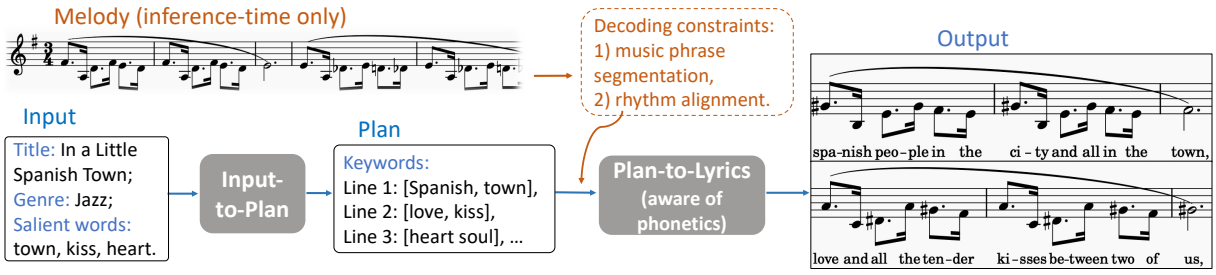


Figure 2: An overview of our approach that disentangles training from inference. Blue parts represent components used during both training and inference, while brown means inference only. During training, our input-to-plan model learns to predict the sentence-level plan (i.e., keywords) given the title, genre, and salient words as input. Then, the plan-to-lyrics model generates the lyrics while being aware of word phonetic information and syllable counts. At inference time, we compile the given melody into 1) music phrase segments and 2) rhythm constraints to guide the generation.

Task Formulation We aim to generate lyrics that comply with both the provided topic and melody. The input topic is further decomposed into an intended title T and a few salient words S to be included in the generated lyrics (see Figure 2 for an example input). Following the settings of previous work (Chen and Lerch, 2020; Sheng et al., 2021), we assume that the input melody $\mathcal{M}$ is predefined and consists of M music phrases ($\mathcal{M} = \{p_1, p_2, ...p_M\}$), and each music phrase contains N_i music notes ($p_i = \{n_{i1}, n_{i2}, ...n_{iN_i}\}$). The output is a piece of lyrics $\mathcal{L}$ that aligns with the music notes: $\mathcal{L} = \{w_{11}, w_{12}, ..., w_{MN}\}$. Here, for $j \in \{1, 2, ..., N_i\}$, w_{ij} is a word or a syllable of a word that aligns with the music note n_{ij}.

3 Lyric Generation Model

We draw inspirations from recent generation models with intermediate outlines as content planning. These models are shown to achieve increased coherence and relevance over end-to-end generation frameworks in other tasks such as story generation (Fan et al., 2018; Yao et al., 2019; Yang et al., 2022). Our lyrics generation model is similarly hierarchical as is shown in Figure 2. Specifically, we finetune two modules in our purely text-based pipeline: 1) an input-to-plan generator that generates a keyword-based intermediate plan, and 2) a plan-to-lyrics generator which is aware of word phonetics and syllable counts.

3.1 Input-to-Plan

In real-world scenarios, users will likely have an intended topic (e.g., a title and a few keywords) to write about. We similarly extract a few salient words from the training lyric using the YAKE algorithm (Campos et al., 2020), and feed them to our input-to-plan module to improve topic relevance.

Model	Output: Generated Lyric
Naïve	Cause the Christmas gift was for.
Chen and Lerch (2020)	Hey now that's what you ever.
Sheng et al. (2021)	Believe you like taught me to.
Ours, Multi-task	Night and day my dreams come true.

Table 1: Examples of lyrics generated by different models with seven syllable counts as a constraint. Our model with multi-task auxiliary learning is the only system that successfully generates a complete line of lyrics with the desired number of syllables. On the other hand, the supervised models (Chen and Lerch, 2020; Sheng et al., 2021) trained with melody-lyrics paired data still generate dangling or cropped lyrics.

The input contains the song title, the music genre, and three salient words extracted from ground truth lyrics. Note that we chose 3 as a reasonable number for practical use cases, but our approach works for any arbitrary number of salient keywords.

Our input-to-plan model is then trained to generate a line-by-line keyword plan of the song. Considering that at inference time we might need different numbers of keywords for different expected output lengths, the number of planned keywords is not fixed. Specifically, we follow the settings used by Tian and Peng (2022) and include a placeholder (the <MASK> token) in the input for every keyword to be generated in the intermediate plan. In this way, we have control over how many keywords we would like per line. We finetune BART-large (Lewis et al., 2020) as our input-to-plan generator with format control.

3.2 Plan-to-Lyrics

Our plan-to-lyrics module takes in the planned keywords as input and generates the lyrics. This module encounters an added challenge: to match the music notes of a given melody at inference time, it should be capable of generating lyrics with a desired syllable count that aligns with the melody.

Task	Sample Data (Input → Output)
T1	Line 1: 8 syllables; Keywords: ... → Line 1: Moon river wider than a mile; ....
T2	Moon river wider than a mile → 8
T3	Line 1: 8 syllables; Keywords: ... → Line 1: Moon (1) river (3) wider (5) than (6) a (7) mile (8); ....
T4	Moon → MUWN; river → RIH_VER; wider → WAY_DER; ...

Table 2: Sample data of the four proposed tasks to facilitate lyric generation with syllable planning.

If we naïvely force the generation to stop once it reaches the desired number of syllables, the outputs are usually cropped abruptly or dangling. For example, if the desired number of syllables is 7, a system unaware of this constraint might generate 'Cause the Christmas gift was for' which is cropped and incomplete. Moreover, two recent lyric generators which are already trained on melody-to-lyrics aligned data also face the same issue (Table 1).

We hence propose to study an under-explored task of *syllable planning*: generating a line of lyrics that 1) is a self-contained phrase and 2) has the desired number of syllables. To this end, we include both the intermediate plan and the desired syllable count as input. Additionally, we propose to equip the plan-to-lyrics module with the word phonetics information and the ability to count syllables. We then adopt multi-task auxiliary learning to incorporate the aforementioned external knowledge during training, as Liebel and Körner (2018); Guo et al. (2019); Poth et al. (2021); Kung et al. (2021) have shown that related *auxiliary tasks* help to boost the system performance on the *target task*. Specifically, we study the collective effect of the following related tasks which could potentially benefit the model to learn the target task:

- T1: Plan to lyrics generation with syllable constraints (the target task)
- T2: Syllable counting: given a sentence, count the number of syllables
- T3: Plan to lyrics generation with granular syllable counting: in the output lyric of T1, append the syllable counts immediately after each word
- T4: Word to phoneme translation

We list the sample data for each task in Table 2. We aggregate training samples from the above tasks, and finetune GPT-2 large (Radford et al., 2019) on different combinations of the four tasks. We show our model's success rate on the target task in Table 3 in Section 6.1.

4 Melody-Guided Inference

In this section, we discuss the procedure to compile a given melody into constraints to guide the decoding at inference time. We start with the most straightforward constraints introduced before: 1) segmentation alignment and 2) rhythm alignment. Note that both melody constraints can be updated without needing to retrain the model.

4.1 Segment Alignment Constraints

The segmentation of music phrases should align with the segmentation of lyrics (Watanabe et al., 2018). Given a melody, we first parse the melody into music phrases, then compute the number of music notes within each music phrase. For example, the first music phrase in Figure 2 consists of 13 music notes, which should be equal to the number of syllables in the corresponding lyric chunk. Without losing generality, we also add variations to this constraint where multiple notes can correspond to one single syllable when we observe such variations in the gold lyrics.

4.2 Rhythm Alignment Constraints

According to Nichols et al., the stress-duration alignment rule hypothesizes that music rhythm should align with lyrics meter. Namely, shorter note durations are more likely to be associated with unstressed syllables. At inference time, we 'translate' a music note to a stressed syllable (denoted by 1) or an unstressed syllable (denoted by 0) by comparing its duration to the average note duration. For example, based on the note durations, the first music phrase in Figure 2 is translated into alternating 1s and 0s, which will be used to guide the inference decoding.

4.3 Phoneme-Constrained Decoding

At each decoding step, we ask the plan-to-lyrics model to generate candidate complete words, instead of subwords, which is the default word piece unit for GPT-2 models. This enables us to retrieve the word phonemes from the CMU pronunciation dictionary (Weide et al., 1998) and identify the resulting syllable stresses. For example, since the phoneme of the word 'Spanish' is 'S PAE1 NIH0 SH', we can derive that it consists of 2 syllables that are stressed and unstressed.

Next, we check if the candidate words satisfy the stress-duration alignment rule. Given a candidate word w_i and the original logit $p(w_i)$ predicted by the plan-to-lyrics model, we introduce a factor α to control the strength:

$$p'(w_i) = \begin{cases} p(w_i), & \text{if } w_i \text{ satisfies rhythm alignment,} \\ \alpha p(w_i), & \text{otherwise.} \end{cases} \quad (1)$$

We can either impose a **hard constraint**, where we reject all those candidates that do not satisfy the rhythm rules ($\alpha = 0$), or impose a **soft constraint**, where we would reduce their sampling probabilities ($0 < \alpha < 1$). Finally, we apply diverse beam search (Vijayakumar et al., 2016) to promote the diversity of the generated sequences.

5 Experimental Setup

In this section, we describe the train and test data, baseline models, and evaluation setup. The evaluation results are reported in Section 6.

5.1 Dataset

Train data. Our training data consists lyrics of 38,000 English songs and their corresponding genres such as Pop, Jazz, and Rock, which we processed from the Genre Classification dataset (Bejan, 2020). The phonetic information needed to construct the auxiliary tasks to facilitate the syllable count control is retrieved from the CMU pronunciation dictionary (Weide et al., 1998).

Automatic test data. The testing setup is the complete diagram shown in Figure 2. Our input contains both the melody (represented in music notes and phases) and the title, topical, and genre information. Our test melodies come from from the lyric-melody aligned dataset (Yu et al., 2021). In total, we gathered 120 songs that do not appear in the training data. Because the provided lyrics are pre-tokenized at the syllable level (e.g. "a lit tle span ish town" instead of "a little spanish town"), we manually reconstructed them back into natural words when necessary.

Two sets of human test data. To facilitate human evaluation, we leverage an online singing voice synthesizer (Hono et al., 2021) to generate the sung audio clips. This synthesizer however requires files in the musicXML format that none of the existing datasets provide (including our automatic test data). Therefore, we manually collected 6 copyrighted popular songs and 14 non-copyrighted public songs from the musescore platform that supports the musicXML format.

The first set of *pilot* eval data are these 20 pieces of melodies that come with ground truth lyrics. In addition, we composed a second, *larger* set of 80 test data by pairing each existing melody with various other user inputs (titles and salient words). This second eval set, which does not come with ground truth lyrics, is aimed at comparison among all the models.

5.2 Baseline Models for Lyrics Generation

We compare the following models. **1. SongMASS** (Sheng et al., 2021) is a state-of-the-art (SOTA) song writing system which leverages masked sequence to sequence pre-training and attention based alignment for M2L generation. It requires melody-lyrics aligned training data while our model does not. **2. GPT-2 finetuned on lyrics** is a uni-modal, melody-unaware GPT-2 large model that is finetuned end-to-end (i.e., **title-to-lyrics**). In the automatic evaluation setting, we also compare an extra variation, **content-to-lyrics**, in which the input contains the title, salient words, and genre. These serve as ablations of the next model LYRA *w/o rhythm* to test the efficacy of our plan-and-write pipeline without inference-time constraints. **3. LYRA w/o rhythm** is our base model consisting of the input-to-plan and plan-to-lyrics modules with segmentation control, but without the rhythm alignment. **4. LYRA w/ soft/hard rhythm** is our multi-modal model with music segmentation and soft or hard rhythm constraints. For the soft constraints setting, the strength controlling hyperparameter $\alpha = 0.01$. All models except SongMASS are finetuned on the same lyrics training data described in Section 5.1.

5.3 Automatic Evaluation Setup

We automatically assess the generated lyrics on two aspects: the quality of text and music alignment. For **text quality**, we divide it into 3 subaspects: 1) **Topic Relevance**, measured by input salient word coverage ratio, and sentence- or corpus-level BLEU (Papineni et al., 2002); 2) **Diversity**, measured by distinct unigrams and bigrams (Li et al., 2016); 3) **Fluency**, measured by the perplexity computed using Huggingface's pretrained GPT-2. We also compute the ratio of cropped sentences among all sentences to assess how well they fit music phrase

segments. For **music alignment**, we compute the percentage where the stress-duration rule holds.

5.4 Human Evaluation Setup

Turker Qualification We used qualification tasks to recruit 120 qualified annotators who 1) have enough knowledge in song and lyric annotation, and 2) pay sufficient attention on the Mechanical Turk platform. The qualification consisted of two parts accordingly. First, to test the Turkers' domain knowledge, we created an annotation task consisting of the first verse from 5 different songs with gold labels. The 5 songs are carefully selected to avoid ambiguous cases, so that the quality can be clearly identified. We selected those whose scores have a high correlation with gold labels. Second, we adopted attention questions to rule out irresponsible workers. As is shown in the example questionnaire in Appendix A, we provided music sheets for each song in the middle of the questions. We asked all annotators the same question: "Do you think the current location where you click to see the music sheet is ideal?". Responsible answers include "Yes" or "No", and suggesting more ideal locations such as "immediately below the audio clip and above all questions". We ruled out irresponsible Turkers who filled in geographical locations (such as country names) in the provided blank.

Annotation Task Our annotation is relative, meaning that annotators assess a group of songs generated from different systems with the same melody and title at once. We evaluated all baseline models except for GPT-2 finetuned (content-to-lyrics), as the two GPT-2 variations showed similar performance in automatic evaluation. We thus only included one due to resource constraints of the human study. Each piece of music was annotated by at least three workers, who were asked to evaluate the quality of the lyrics using a 1-5 Likert scale on six dimensions across musicality and text quality. For musicality, we asked them to rate **singability** (whether the melody's rhythm aligned well with the lyric's rhythm) and **intelligibility** (whether the lyric content was easy to understand when listened to without looking at the lyrics).[3] For the lyric quality, we asked them to rate **coherence**, **creativeness**, and **in rhyme**. Finally, we asked annotators to rate how much they liked the song **overall**. A complete example of the survey can be found in Appendix A. The workers were paid $16 per hour and the average inter-annotator agreement in terms of Pearson correlation was 0.47.

[3]The task was carefully designed so that intelligibility was asked before the workers read the lyrics. See Appendix A for more details.

Task Name				Success Rate	
T1 Lyrics	T2 Count	T3 Granular	T4 Phoneme	Greedy Decode	Sampling Decode
✓				23.14%	19.87%
✓	✓			50.14%	44.64%
	✓	✓		55.01%	49.70%
✓	✓	✓		**93.60%**	**89.13%**
✓	✓	✓	✓	91.37%	87.65%

Table 3: Success rate for variants of our plan-to-lyrics model on generating sentences with the desired number of syllables.

6 Results

6.1 Generating a Sequence of Lyrics with the Desired Number of Syllables

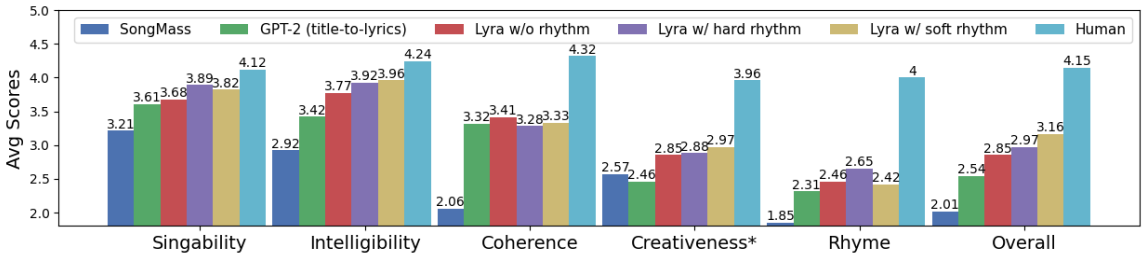
Recall that in Section 3.2, we trained the plan-to-lyrics generator on multiple auxiliary tasks in order to equip it with the ability to generate a sentence with a pre-defined number of syllables. A sample output (boldfaced) can be found below: *Line 1 (8 syllables):* ***Last Christmas I gave you my gift****; Line 2 (13 syllables):* ***It was some toys and some clothes that I said goodbye to****; Line 3 (11 syllables):* ***But someday the tree is grown with other memories****; Line 4 (7 syllables):* ***Santa can hear us singing****;...*

To test this feature, we compute the average success rate on a held-out set from the training data that contains 168 songs with 672 lines of lyrics. For each test sample, we compute its *success* as a binary indicator where 1 indicates the output sequence contains exactly the same number of syllables as desired, and 0 for all other cases. We experimented with both greedy decoding and sampling, and found that BART (Lewis et al., 2020) could not learn these multi-tasks as well as the GPT-2 family under the same settings. We hence report the best result of finetuning GPT-2 large (Radford et al., 2019) in Table 3.

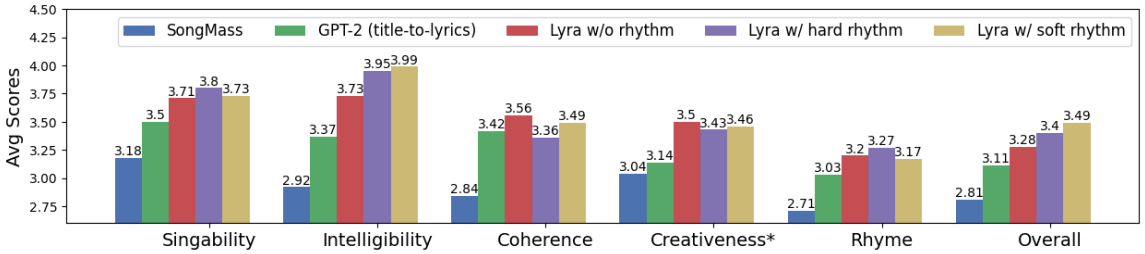
The first row in Table 3 shows that the model success rate is around 20% without multi-task learning, which is far from ideal. By gradually training with auxiliary tasks such as syllable counts, the success rate increases, reaching over 90% (rows 2, 3, 4). This shows the efficacy of multi-task auxiliary learning. We also notice that the phoneme translation task is not helpful for our goal (row 4), so we disregard the last task and only keep the remaining

Model Name	Content Control/Topic Relevance			Diversity		Fluency		Music
	Salient Word Coverage↑	Sent Bleu↑	Corpus Bleu↑	Dist-1↑	Dist-2↑	PPL ↓	Cropped Sentence↓	Stress-Duration
SongMASS (Sheng et al., 2021)	/	0.045	0.006	**0.17**	**0.57**	518	34.51%	58.8%
GPT-2 (title-to-lyrics)	/	0.026	0.020	0.09	0.31	**82**	/	53.6%
GPT-2 (content-to-lyrics)	83.3%	0.049	0.027	0.10	0.42	87	/	54.2%
LYRA w/o rhythm	**91.8%**	0.074	0.046	0.12	0.45	85	**3.65%**	63.1%
LYRA w/ soft rhythm	89.4%	**0.075**	**0.047**	0.11	0.46	85	8.96%	68.4%
LYRA w/ hard rhythm	88.7%	0.071	0.042	0.12	0.45	108	10.26%	**89.5%**
Ground Truth	100%	1.000	1.000	0.14	0.58	93	3.92%	73.3%

Table 4: Automatic evaluation results. Human (ground truth) performance is highlighted in a grey background. Among all models, we highlight the best scores in boldface and underline the second best.



(a) Human evaluation results on the pilot test set with human as ground truth lyrics.



(b) Human evaluation results on the larger test set without ground truth lyrics.

Figure 3: Average human Likert scores for two lyrics evaluation datasets on singability, intelligibility, coherence, creativity, rhyme, and overall quality. For each pair of systems in either study, we conduct paired t-test and observe statistical significance across all dimensions except creativeness (denoted by *).

three tasks in our final implementation (row 3).

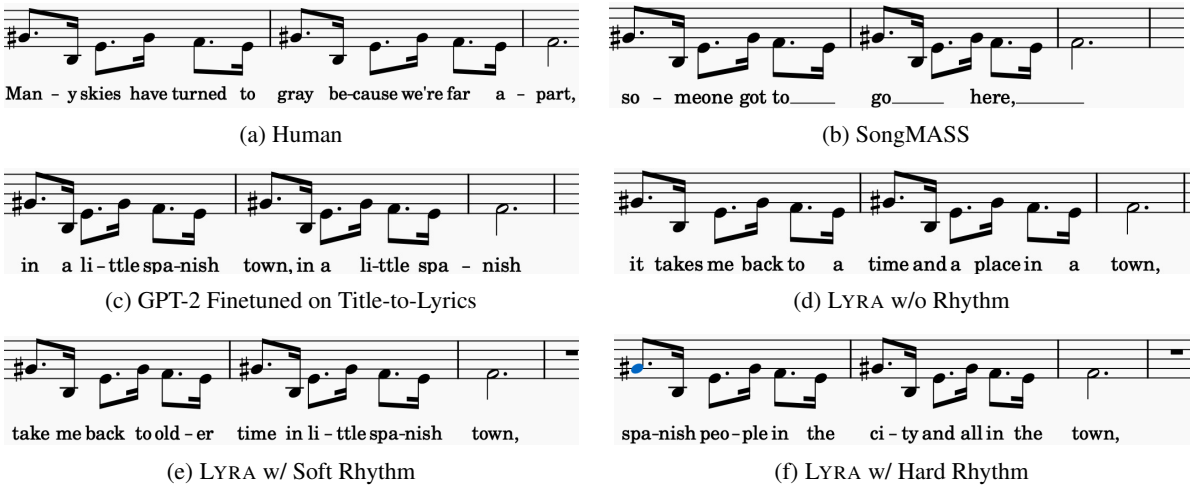
6.2 Automatic Evaluation Results

We report the automatic evaluation results in Table 4. Our LYRA models significantly outperform the baselines and generate the most on-topic and fluent lyrics. In addition, adding rhythm constraints to the base LYRA noticeably increases the music alignment quality without sacrificing too much text quality. It is also noteworthy that humans do not consistently follow stress-duration alignment, meaning that higher is not necessarily better for music alignment percentage. The comparisons between GPT-2 content-to-lyrics and LYRA w/o rhythm support the hypothesis of the better topic control provided by our hierarchical architecture.

Since the baseline model SongMASS has no control over the content, it has lowest topic relevance scores. Moreover, although the SongMASS baseline seems to achieve the best diversity, it tends to produce non-sensical sentences that consist of a few gibberish words (e.g., 'for hanwn to stay with him when, he got to faney he alone'), partially because its training data are pre-tokenized at the syllable level. Such degeneration is also reflected by the extremely high perplexity and cropped sentence ratio (CSR). Meanwhile, CSR is not applicable to both GPT-2 finetuned models because they are melody-unaware and generate lyrics freely without being forced to end at the end of each music segment.

6.3 Human Evaluation Results

The results on both evaluation sets are shown in Figures 3a and 3b. Clearly, human-written lyrics greatly outperform all models. For both evaluation sets, we notice the relative rankings of the models remain the same across all metrics except creativeness. This observation is mirrored by paired t-tests where we find that the best machine model differentiates from the second best machine model with statistical significance (p-value < 0.05) for all aspects except creativeness. Both indicate the reliability



(a) Human (b) SongMASS

(c) GPT-2 Finetuned on Title-to-Lyrics (d) LYRA w/o Rhythm

(e) LYRA w/ Soft Rhythm (f) LYRA w/ Hard Rhythm

Figure 4: Music sheets showing the lyric generated by different systems given the same piece of melody. LYRA with soft and hard rhythm control are the only two models that can generate highly singable lyrics. The singing voices of the complete song can be found in this demo page.

a - Human	**b - SongMASS** (Sheng et al., 2021)
Many skies have turned to gray because we're far apart, Many moons have passed away and still she's in my heart, We made a promise and sealed it with a kiss, In a little Spanish town twas on a night like this.	Someone got to go here, Forget that rest of my life, Everybody loves somebody who i, In the middle of the night when the.
c - GPT-2 Finetuned on Title-to-Lyrics	**d - LYRA w/o Rhythm**
In a little Spanish town, In a little Spanish town, We'll make you feel good, We'll make you dance, In a little Spanish town, In a little Spanish town . . .	It takes me back to a time and a place in a *town*, But if you do *love* me you will give me some *kisses*, Where my *heart* and *soul* are not the same but now, Your heart is in my *heart* and not in a *soul* i know.
e - LYRA w/ Soft Rhythm	**f - LYRA w/ Hard Rhythm**
Take me back to older time in little *Spanish town*, And all the *love* and all the *kisses* that you gave me, I need your *heart* and your *soul* and your love too, I need your *heart* and *soul* and I need you back again.	*Spanish* people in the city and all in the *town*, *Love* and all the tender *kisses* between two of us, Is it my *heart* or my *soul* in you and me, In the *heart* and in the *soul* and in the mind and then.

Table 5: An example of the generated lyrics with the same input - Title: In a little Spanish Town; Genre: Jazz; Salient words: town, kisses, heart. We highlight the generated keywords in italics.

of our collected results in singability, intelligibility, coherence, rhyme, and overall quality.

LYRA with hard or soft rhythm constraint are the best models in terms of singability, intelligibility, rhyme, and overall quality, which demonstrates the efficacy of our plan-and-write pipeline with melody alignment. We regard LYRA with soft rhythm as our best model since it has highest overall quality. The addition of soft rhythm alignment leads to further improvements in musicality and overall quality, with only a little sacrifice in coherence compared to GPT-2 (title-to lyrics). On the other hand, imposing hard rhythm constraints sacrifices the coherence and intelligibility of lyrics.

Surprisingly, SongMASS performs even worse than the finetuned GPT-2 baseline in terms of musicality. Upon further inspection, we posit that SongMASS too often deviates from common singing habits: it either assigns two or more syllables to one music note, or matches one syllable with three or more consecutive music notes.

6.4 Qualitative Analysis

We conduct a case study on an example set of generated lyrics to better understand the advantages of our model over the baselines. In this example, all models generate lyrics given the same title, genre, and salient words, as well as the melody of the original song. We show the music sheet of the first generated segment in Figure 4 and the complete generated lyrics in Table 5. We also provide the song clips with synthesized singing voices and more examples in this demo website.

Musicality. The melody-lyric alignment in Figure 4 is representative in depicting the pros and cons of the compared models. Although SongMASS is supervised on parallel data, it still often assigns too many music notes to one single syllable, which reduces singability and intelligibility. The GPT-2 title-to-lyrics model is not aware of the melody and thus fails to match the segmentation of music phrase with the generated lyrics.

LYRA w/o rhythm successfully matches the segments, yet stressed and long vowels such as in the words 'takes' and 'place' are wrongly mapped to short notes. Humans, as well as our models with both soft and hard rhythm alignment, produce singable lyrics.

Text quality. As shown in Table 5, SongMASS tends to generate simple and incoherent lyrics because it is trained from scratch. The GPT-2 title-to-lyrics model generates coherently and fluently, but is sometimes prone to repetition. All three variations of LYRA benefit from the hierarchical planning stage and generate coherent and more informative lyrics. However, there is always a **trade-off between musicality and text quality**. Imposing hard rhythm constraints could sometimes sacrifice coherence and creativity and thus hurt the overall quality of lyrics.

7 Related Work

7.1 Melody Constrained Lyrics Generation

End-to-End Models. Most existing works on M2L generation are purely data-driven and suffer from a lack of aligned data. For example,Watanabe et al. (2018); Lee et al. (2019); Chen and Lerch (2020) naively apply SeqGAN (Yu et al., 2017) or RNNs to sentence-level M2L generation. The data collection process is hard to automate and leads to manual collection of only small amounts of samples. Recently, Sheng et al. (2021) propose SongMASS by training two separate transformer-based models for lyric or melody with cross attention. To the best of our knowledge, our model LYRA is the first M2L generator that does not require any paired cross-modal data, and is trained on a readily available uni-modal lyrics dataset.

Integrating External Knowledge. Oliveira et al. (2007); Oliveira (2015) apply rule-based text generation methods with predefined templates and databases for Portuguese. Ma et al. (2021) use syllable alignments as reward for the lyric generator. However, it only estimates the expected number of syllables from the melody. We not only provide a more efficient solution to syllable planning, but also go one step further to incorporate the melody's rhythm patterns by following music theories (Nichols et al., 2009; Dzhambazov et al., 2017). Concurrently, Xue et al. (2021); Guo et al. (2022) partially share similar ideas with ours and leverage the sound to generate Chinese raps or translate lyrics via alignment constraints. Nevertheless, the phonetics of Chinese characters are very different from English words, and rap generation or translation is unlike M2L generation.

7.2 NLG with Hierarchical Planning

Hierarchical generation frameworks are shown to improve consistency over sequence-to-sequence frameworks in other creative writing tasks such as story generation (Fan et al., 2018; Yao et al., 2019). Recently, a similar planning-based scheme is adopted to poetry generation (Tian and Peng, 2022) to circumvent the lack of poetry data. We similarly equip LYRA with the ability to comply with a provided topic via such content planning.

7.3 Studies on Melody-Lyrics Correlation

Music information researchers have found that it is the duration of music notes, not the pitch values that a play significant role in melody-lyric alignment (Nichols et al., 2009; Dzhambazov et al., 2017). Most intuitively, one music note should not align with two or more syllables, and the segmentation of lyrics should match the segmentation of music phrases for singability and breathability (Watanabe et al., 2018). In addition, Nichols et al. (2009) find out that there is a correlation between syllable stresses and note durations for better singing rhythm. Despite the intuitiveness of the aforementioned alignments, our experiments show that existing lyric generators which are already trained on melody-lyrics aligned data still tend to ignore these fundamental rules and generate songs with less singability.

8 Conclusion and Future Work

Our work explores the potential of lyrics generation without training on lyrics-melody aligned data. To this end, we design a hierarchical plan-and-write framework that disentangles training from inference. At inference time, we compile the given melody into music phrase segments and rhythm constraints. Evaluation results show that our model can generate high-quality lyrics that significantly outperform the baselines. Future directions include investigating more ways to compile melody into constraints such as the beat, tone or pitch variations, and generating longer sequences of lyrics with song structures such as verse, chorus, and bridge. Future works may also take into account different factors in relation to the melody such as mood and theme.

Acknowledgements

The authors would like to thank Yiwen Chen from University of Cambridge for helping to curate evaluation data, as well as designing and providing valuable insights to the human evaluation guidelines. We also thank the anonymous reviewers for the helpful comments.

Limitations

We discuss the limitations of our work. First of all, our model LYRA is build upon pre-trained language models (PTLM) including Bart (Lewis et al., 2020) and GPT-2 (Radford et al., 2019). Although our method is much more data friendly than previous methods in that it does not require training on melody-lyric aligned data, our pipeline may not apply to low-resource languages which do not have PTLMs. Second, our current adoption of melody constraints is still simple and based on a strong assumption of syllable stress and note duration. We encourage future investigation about other alignments such as the tone or pitch variations. Lastly, although we already have the music genre as an input feature, it remains an open question how to analyze or evaluate the generated lyrics with respect to a specific music genre.

Ethics Statement

It is known that the generated results by PTLMs could capture the bias reflected in the training data (Sheng et al., 2019; Wallace et al., 2019). Our models may potentially generate offensive content for certain groups or individuals. We suggest to carefully examine the potential biases before deploying the models to real-world applications.

References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2019. An effective approach to unsupervised machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 194–203.

Matei Bejan. 2020. Multi-lingual lyrics for genre classification.

Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Jorge, Célia Nunes, and Adam Jatowt. 2020. Yake! keyword extraction from single documents using multiple local features. *Information Sciences*, 509:257–289.

Yihao Chen and Alexander Lerch. 2020. Melody-conditioned lyrics generation with seqgans. In *2020 IEEE International Symposium on Multimedia (ISM)*, pages 189–196. IEEE.

Georgi Dzhambazov et al. 2017. *Knowledge-based probabilistic modeling for tracking lyrics in music audio signals*. Ph.D. thesis, Universitat Pompeu Fabra.

Tim Edensor. 2020. *National identity, popular culture and everyday life*. Routledge.

Darren Edmonds and Joao Sedoc. 2021. Multi-emotion classification for song lyrics. In *Proceedings of the Eleventh Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 221–235.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

Angela Fan, Mike Lewis, and Yann Dauphin. 2019. Strategies for structuring story generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2650–2660, Florence, Italy. Association for Computational Linguistics.

Fenfei Guo, Chen Zhang, Zhirui Zhang, Qixin He, Kejun Zhang, Jun Xie, and Jordan Boyd-Graber. 2022. Automatic song translation for tonal languages. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 729–743, Dublin, Ireland. Association for Computational Linguistics.

Han Guo, Ramakanth Pasunuru, and Mohit Bansal. 2019. Autosem: Automatic task selection and mixing in multi-task learning. *arXiv preprint arXiv:1904.04153*.

Yukiya Hono, Kei Hashimoto, Keiichiro Oura, Yoshihiko Nankaku, and Keiichi Tokuda. 2021. Sinsy: A deep neural network-based singing voice synthesis system. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:2803–2815.

Po-Nien Kung, Sheng-Siang Yin, Yi-Cheng Chen, Tse-Hsuan Yang, and Yun-Nung Chen. 2021. Efficient multi-task auxiliary learning: Selecting auxiliary data by feature similarity. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 416–428, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. Unsupervised machine translation using monolingual corpora only. In *International Conference on Learning Representations*.

Hsin-Pei Lee, Jhih-Sheng Fang, and Wei-Yun Ma. 2019. icomposer: An automatic songwriting system for chinese popular music. In *Proceedings of the 2019 Conference of the North American Chapter of the*

Association for Computational Linguistics (Demonstrations), pages 84–88.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics,*.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Lukas Liebel and Marco Körner. 2018. Auxiliary tasks in multi-task learning. *arXiv preprint arXiv:1805.06334*.

Xichu Ma, Ye Wang, Min-Yen Kan, and Wee Sun Lee. 2021. Ai-lyricist: Generating music and vocabulary constrained lyrics. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 1002–1011.

Eric Nichols, Dan Morris, Sumit Basu, and Christopher Raphael. 2009. Relationships between lyrics and melody in popular music. In *ISMIR 2009-Proceedings of the 11th International Society for Music Information Retrieval Conference*, pages 471–476.

Hugo Gonçalo Oliveira. 2015. Tra-la-lyrics 2.0: Automatic generation of song lyrics on a semantic domain. *Journal of Artificial General Intelligence*, 6(1):87.

Hugo R Gonçalo Oliveira, F Amilcar Cardoso, and Francisco C Pereira. 2007. Tra-la-lyrics: An approach to generate text based on rhythm. In *Proceedings of the 4th. International Joint Workshop on Computational Creativity*. A. Cardoso and G. Wiggins.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Clifton Poth, Jonas Pfeiffer, Andreas Rücklé, and Iryna Gurevych. 2021. What to pre-train on? efficient intermediate task selection. *arXiv preprint arXiv:2104.08247*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Emily Sheng, Kai-Wei Chang, Premkumar Natarajan, and Nanyun Peng. 2019. The woman worked as a babysitter: On biases in language generation. *arXiv preprint arXiv:1909.01326*.

Zhonghao Sheng, Kaitao Song, Xu Tan, Yi Ren, Wei Ye, Shikun Zhang, and Tao Qin. 2021. Songmass: Automatic song writing with pre-training and alignment constraint. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13798–13805.

Xu Tan and Xiaobing Li. 2021. A tutorial on ai music composition. In *Proceedings of the 29th ACM international conference on multimedia*, pages 5678–5680.

Yufei Tian and Nanyun Peng. 2022. Zero-shot sonnet generation with discourse-level planning and aesthetics features. *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Alexandros Tsaptsinos. 2017. Lyrics-based music genre classification using a hierarchical attention network. *arXiv preprint arXiv:1707.04678*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2016. Diverse beam search: Decoding diverse solutions from neural sequence models. *arXiv preprint arXiv:1610.02424*.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing nlp. *arXiv preprint arXiv:1908.07125*.

Kento Watanabe, Yuichiroh Matsubayashi, Satoru Fukayama, Masataka Goto, Kentaro Inui, and Tomoyasu Nakano. 2018. A melody-conditioned lyrics language model. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 163–172.

Robert Weide et al. 1998. The carnegie mellon pronouncing dictionary. *release 0.6, www. cs. cmu. edu*.

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. *arXiv preprint arXiv:2107.01875*.

Kevin Yang, Nanyun Peng, Yuandong Tian, and Dan Klein. 2022. Re3: Generating longer stories with recursive reprompting and revision. *Empirical Methods in Natural Language Processing*.

Lili Yao, Nanyun Peng, Ralph Weischedel, Kevin Knight, Dongyan Zhao, and Rui Yan. 2019. Plan-and-write: Towards better automatic storytelling. In

Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, pages 7378–7385.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

Yi Yu, Abhishek Srivastava, and Simon Canales. 2021. Conditional lstm-gan for melody generation from lyrics. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 17(1):1–20.

Chen Zhang, Luchin Chang, Songruoyao Wu, Xu Tan, Tao Qin, Tie-Yan Liu, and Kejun Zhang. 2022. Relyme: Improving lyric-to-melody generation by incorporating lyric-melody relationships. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 1047–1056.

A Survey Form Used In Human Evaluation

We show the original survey with the evaluation instructions and the annotation task in Figures 5 through 9. Figure 5, Figure 6, and Figure 7 provide task instructions, including the definition of each metric (Intelligibility, Singability, Coherence, Creativeness, and Rhyme), and examples of good and bad lyrics in each criterion. Figures 8 and 9 showcase the actual annotation task.

In the the actual annotation tasks, we noticed that annotators tended to adjust their rating to *Intelligibility* (whether the content of the lyrics was easy to understand *without looking at the lyrics*) after they were prompted to see the lyrics texts. We hence explicitly asked them to rate *Intelligibility* twice, both before and after they saw the generated lyrics and music scores. Annotators must not modify their ratings to the first question after they saw the lyric texts, but could still use the second question to adjust their scores if needed. Such a mechanism helped us reduce the noise introduced by the presentation of lyric texts and music sheets. Namely, we asked the same questions twice, but only took into account the first intelligibility ratings when we computed the results.

Task Instructions

In the survey, you will be given audio clips (with music sheet that is just for reference in case the synthesized audios are bad). Each of them is a verse of a song lyrics. For each verse, your job is to evaluate the lyrics on 5 criteria (click to see definition):

▼ Intelligibility

1. Intelligibilty is whether the content of the lyrics is easy to understand *without looking at the lyrics*. A higher score means the lyrics is easier to understand, while a lower score indicates the likelihood to mishear the lyrics. For more details, please refer to the examples below.

▼ Singability

1. Singability is what makes a song lyrics easier to sing. A higher score means means *the melody's rhythm aligns well with the lyric's rhythm when it is spoken as a natural conversation*. A low score indicates the reverse, e.g. one single music note corresponding to many syllables in the lyrics, and/or a long and pronouced music note corresponding to an unstressed syllable. For more details, please refer to the examples below.

▼ Coherence

1. Coherence is whether the quality of the lyrics is logical and consistent as a whole.

▼ Creativeness

1. Creativeness is whether the lyrics content surprises you in a good way. For example, lyrics with figurative languages such as similes and metaphors are more creative.

▼ Rhyme

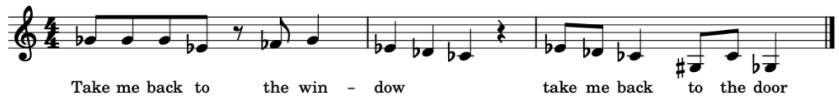
1. Rhyme is whether the lyrics sound in rhyme.

We would like to show you a few examples of good and bad lyrics in each criterion.

Beware that we use a singing synthesizer to sing lyrics, which might mispronounce the lyrics. Please try not being distracted by the mistakes of synthesizer.

Lyrics 1 with good singability, intelligibility and rhyme

0:00 / 0:12



Lyrics: *take me back to the window, take me back to the door.*

The rhythm of the lyrics is close to our natural conversation, which makes the lyrics easy to sing.

The lyrics is easy to understand when we first hear it, which makes the lyrics with good intelligibility.

The last syllable of each line sound similar in the lyrics, which means the lyrics is in rhyme.

Therefore, the lyrics achieves **5** in singability, intelligibility and rhyme.

The lyrics makes sense to us, so it achieves **5** in coherence as well.

The content does not surprises us, so it achieves **3** in creativeness.

Lyrics 2 with good creativeness

0:00 / 0:20

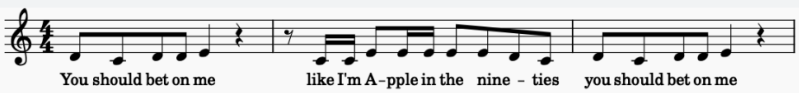
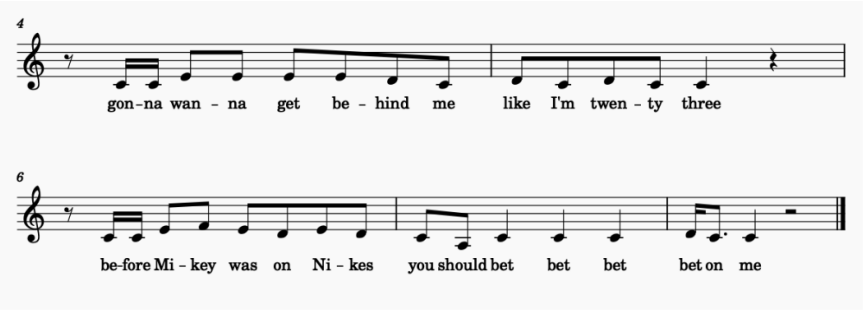


Figure 5: Task Instruction Page 1

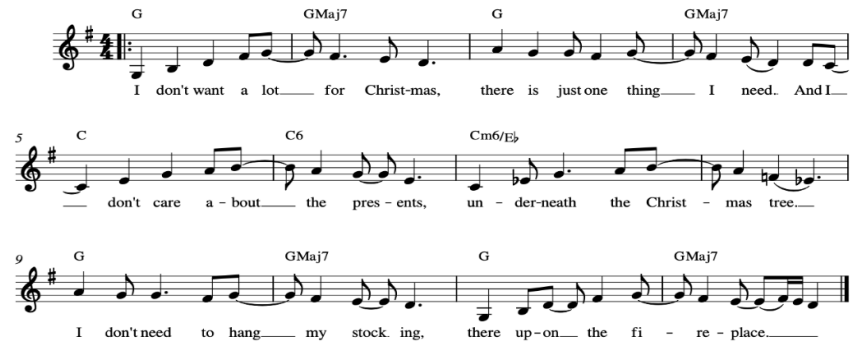


Lyrics: *you should bet on me, like I'm Apple in the '90s, you should bet on me, gonna wanna get behind me, like I'm 23, before Mikey was on Nikes, you should bet, bet, bet, bet on me.*

The lyrics is creative because it uses similes (the comparison of one thing with another thing of a different kind). "I" am compared to Apple and Mikey, which shows my potential, therefore the lyrics achieves **5** in creativeness.

Lyrics 3 with good coherence

0:00 / 0:33

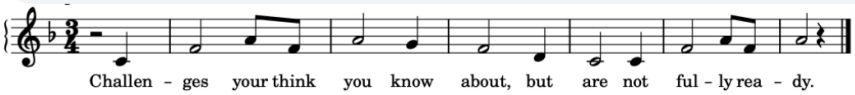


Lyrics: *I don't want a lot for Christmas, there is just one thing I need. And I don't care about the presents, underneath the Christmas tree. I dont' need to hang my stocking, there upon the fireplace*

The lyrics is coherent as the its content focus on a story of Christmas, therefore it achieves **5** in coherence.

Lyrics 4 with bad singability

0:00 / 0:18



Lyrics: *Challenges your think you know about, but are not fully ready.*

The lyrics is difficult to sing with it as its rhythm strongly violates the rhythm when the sentence is spoken as a natural conversation.

For example, the word '*challenges*' is spoken with an accent on the first part ('*cha*') and an unstress on the last part ('*ges*'), yet the melody has a long and strong note corresponding to unstressed part ('*ges*'), making it akward to sing.

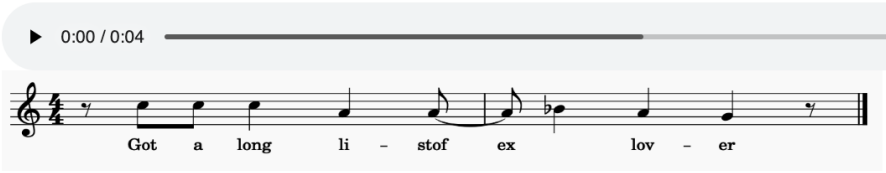
In addition, there are cases where two syllables ('*cha-llenge*' and '*a-bout*') that correspond with one music note, so the singer has to sing them in a hurry. Therefore the whole piece achieves 1 in singability.

Lyrics 5 with bad intelligibility

0:00 / 0:04

Figure 6: Task Instruction Page 2

Lyrics 5 with bad intelligibility

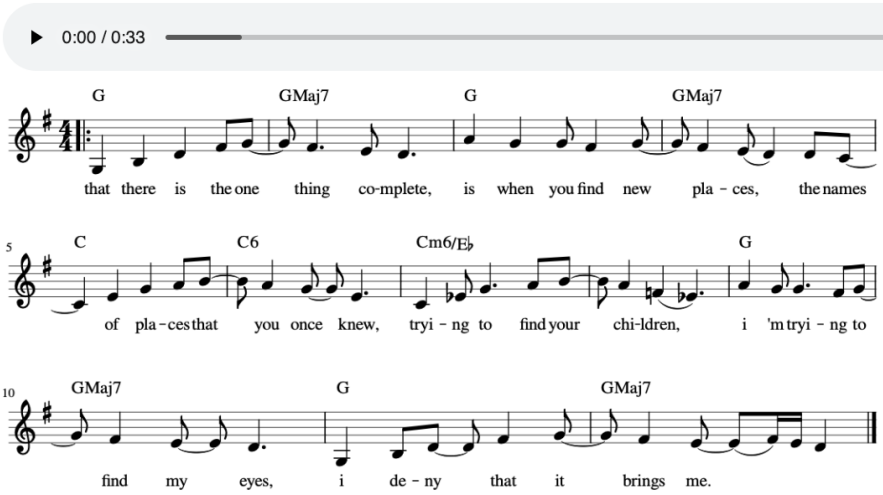


Correct lyrics: *got a long list of ex-lovers.*

Misheard lyrics: *all the lonely Starbucks lovers.*

The misunderstanding of the lyrics content when we hear the song means the lyrics is bad in intelligibility,therefore, it achieves **1** in intelligibility.

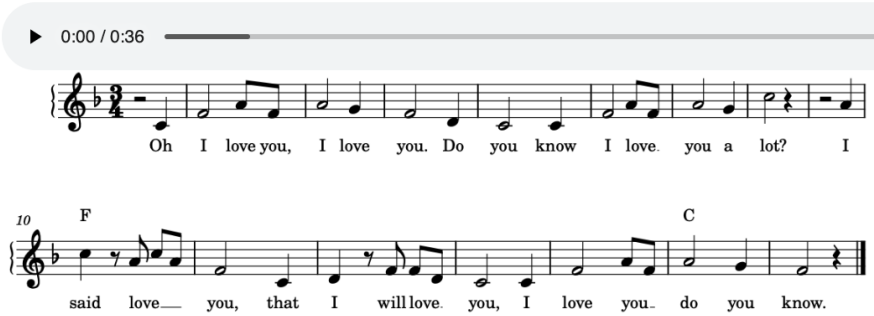
Lyrics 6 with bad coherence



Lyrics: *that there is the one thing complete, is when you find new places, the names of places that you once knew, trying to find your children, I'm trying to find my eyes, I deny that it brings me.*

The lyrics is difficult to understand what it means because of grammar errors and the lack of main plot, therefore it achieves **1** in coherence.

Lyrics 7 with bad creativeness



Lyrics: *Oh I love you, I love you. Do you know I love you a lot? I said love you, that I will love you, I love you do you know.*

The lyrics keeps repeated, which does not bring us much new information, therefore, it achieves **1** in creativeness.

Figure 7: Task Instruction Page 3

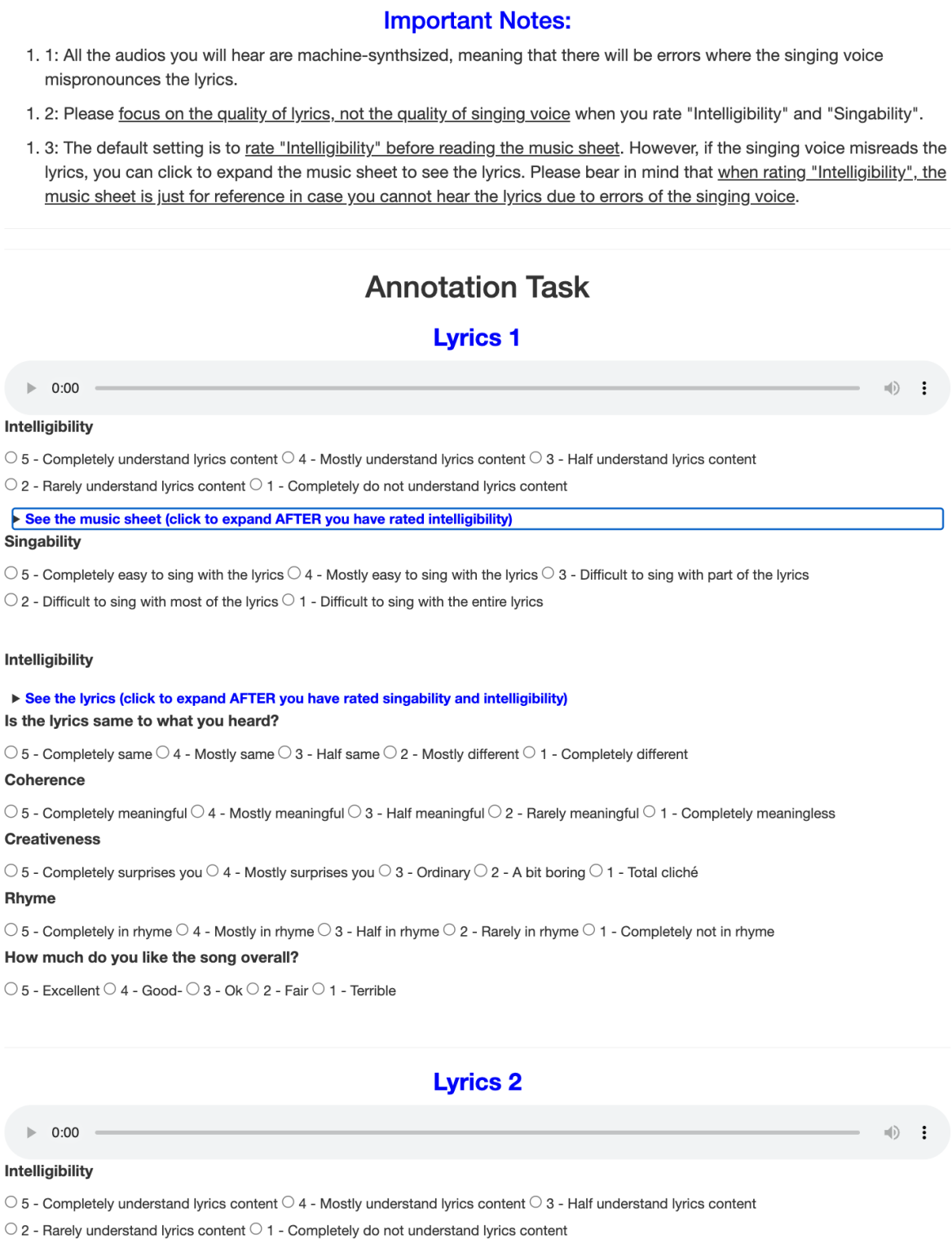


Figure 8: Annotation Task Page 1. We explicitly asked the annotators to rate Intelligibility twice, before and after they saw the generated lyrics and provided musicality scores. See explanations in Appendix A.

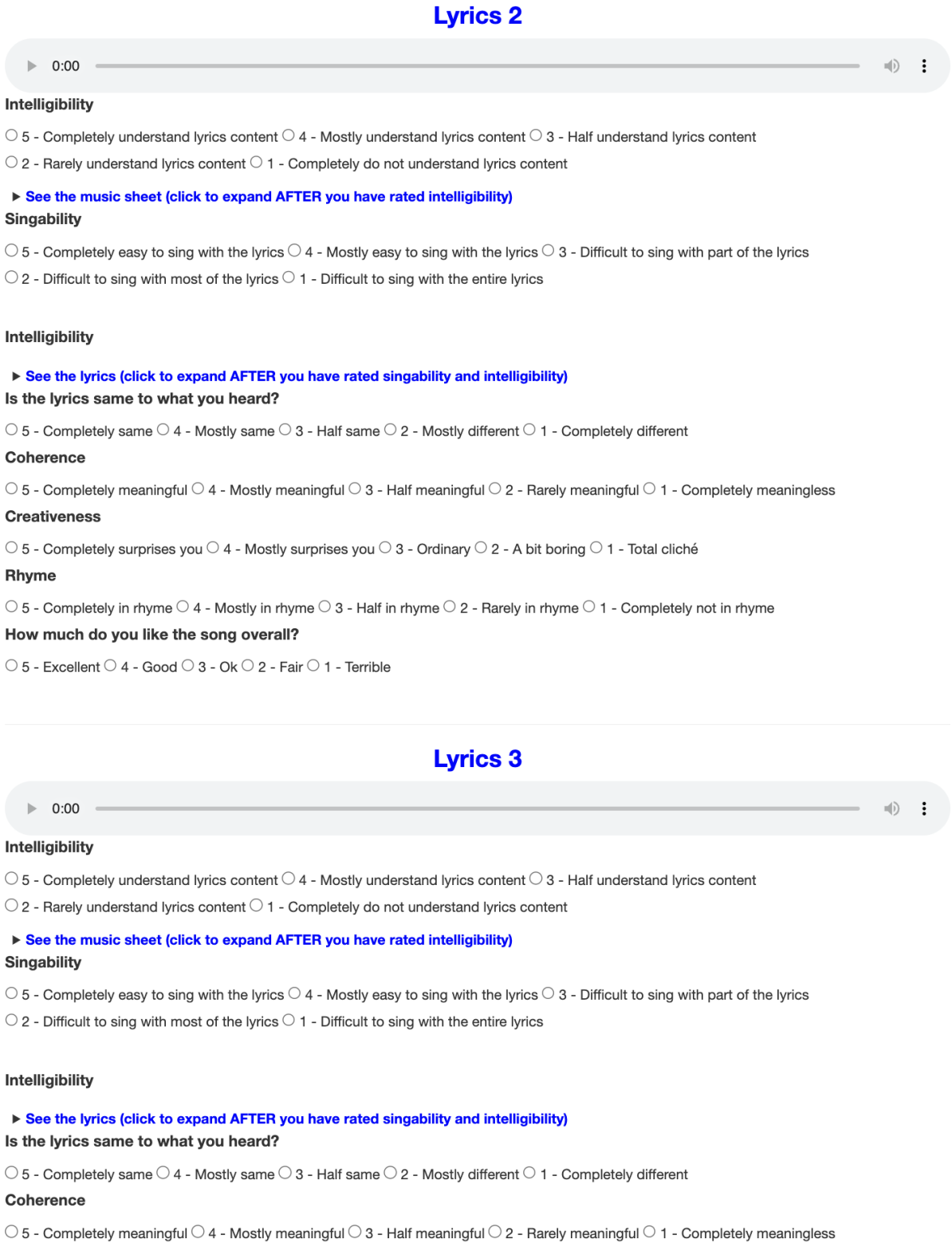


Figure 9: Annotation Task Page 2. We explicitly asked the annotators to rate Intelligibility twice, before and after they saw the generated lyrics and provided musicality scores.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Left blank.

☑ A2. Did you discuss any potential risks of your work?
Left blank.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Left blank.

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☒ Did you use or create scientific artifacts?

Left blank.

☑ B1. Did you cite the creators of artifacts you used?
Left blank.

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
2.1

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
5.1

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Left blank.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Left blank.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Left blank.

C ☑ Did you run computational experiments?

5

☒ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Left blank.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Left blank.

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Left blank.

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Left blank.

D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Not applicable. Left blank.

☑ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Left blank.

☑ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Left blank.

☑ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.