Scansion-based Lyrics Generation

Yiwen Chen, Simone Teufel
Department of Computer Science and Technology
University of Cambridge
United Kingdom
{yc429, sht25}@cam.ac.uk

Abstract

We aim to generate lyrics for Mandarin songs with a good match between the melody and the tonal contour of the lyrics. Our solution relies on mBart, treating lyrics generation as a translation problem, but rather than translating directly from the melody as is common, we generate from scansion as an intermediate contour representation that can fit a given melody. One of the advantages of our solution is that it does not require a parallel melody-lyrics dataset. We also present a thorough automatic evaluation of our system against competitors, using several new evaluation metrics. These measure intelligibility, fit to melody, and use proxies for quantifying creativity (variation, semantic similarity to keywords, and perplexity). We compare different implementations of scansion to competitor systems. Our best system outperforms all others in lyric-melody fit and is in the top group of systems for two of the creativity metrics (variation and perplexity), overshadowing two large language models (LLM) specialised to this task.

Keywords: lyrics generation, creativity, tone-melody match, evaluation metrics

1. Introduction

Some lyrics are easy to sing and to remember because the words follow the melody. We aim to generate lyrics of this type. Lyrics generation is often modelled on poetry generation, but it comes with an additional challenge: the number of syllables required is variable as it depends on the number of notes in the song, in contrast to poem generation, where a fixed number of syllables is prescribed. End-to-end lyrics generation architectures often find it hard to produce the correct number of syllables, as well having problems with melody–lyrics fit; additionally, they require a large parallel lyrics-melody dataset.

Our approach, which takes phonetic knowledge and songwriting theory into consideration, is based on the concept of scansion. Scansion is a graphical analysis method for deciding the stress pattern of words in lyrics or poetry. It is widely used in classical poetry writing (Greene and Cushman, 2016). A poet selects a metrical pattern before composition. For example, iambic pentameter represents a metrical pattern consisting of five feet per line, where each foot contains an unstressed syllable followed by a stressed syllable. In return, when people read a poem it is scansion that allows them to identify the poem's metrical pattern.

Consider the following example of scansion, representing the first line in *When I Consider How My Light Is Spent* by John Milton (1608-74).

×	/	×	/	×	/	×	/	×	/
When	I |	con	sid |	er	how |	my	light |	is	spent

Here, “ / ” and “ x ” represent stressed and unstressed syllables respectively; lines are segmented into feet[1] by vertical bars “ | ”.

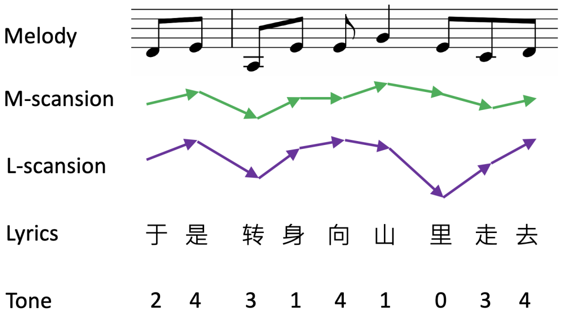

Figure 1: An example of lyrics-melody matching in a Mandarin song.

We are the first to expand the concept of scansion to tonal languages such as Mandarin and Cantonese. This enables us to generate Mandarin lyrics that have a provably better match between melody and tone contour. We do this by correlating two notions of scansion: one representing melody (called M-Scansion) and one representing lyrics (called L-Scansion).

Early Mandarin lyricists paid little attention to aligning tone contours to melodic contours, maybe because Chinese pop music emerged much earlier than the adoption of Mandarin as an official language in 1956 (You and Li, 2015; Yu, 2007). Although many of today's Mandarin lyrics do not exhibit a good match to the contour of their melodies, a rising number of young songwriters produce a new style of lyrics that reverses this tradition by fitting lyrics more closely to the melody (Wee, 2007).

[1] Feet are the fundamental units of poetry composition.

LREC-COLING 2024, pages 14370–14381
20-25 May, 2024.

Close contour matching between the melody and the pitch of the tones also has the advantage of avoiding Mondegreen, a phenomenon where parts of a lyrics are misheard and misinterpreted as a near-homophone with a different meaning. An example of a good lyrics-melody match is given in Fig.1. Contour of melody (M-scansion) is indicated by green arrows; the change of relative pitch of tone of Hanzi (L-scansion) is indicated by purple arrows.

Except intelligibility and good lyrics–melody fit, there are other positive properties we would like our lyrics to have. The first of these is that lyrics should not be boring, cliched or predictable. Searching for objective metrics for this notion, we present three proxy metrics. When the lyrics generator operates on a set of melodies, it should produce a set of lyrics which display some internal variation from each other. The lyrics should also "respond" to keywords, which are given to our system along with the melody, as is common practice in lyrics and poetry generation. The purpose of using such keywords is to control the atmosphere or general theme of the resulting lyrics. We therefore also present a metric of semantic similarity between the keywords and the resulting lyrics. And as a general proxy for creativity, one might also employ information-theoretic metrics of surprise.

In this paper, our contributions are fourfold.

- First, we present a scansion-based lyrics generator that outputs lyrics according to given melodies without requiring a parallel melody-lyrics dataset. M-scansion is calculated on the melodies, resulting in a pseudo melody that can be input to a fine-tuned mBART model. Internally, fine-tuning happens with a parallel dataset which we created from lyrics alone, as these are plentiful.

- Our method of creating this parallel corpus also relies on scansion and is our second contribution. We create the parallel dataset of pseudo melodies and lyrics by applying L-scansion to pre-existing lyrics to finetune a mBART model.

- We test four M-scansion methods based on neumes, cosine similarity, Hidden Markov Model and GPT2. The latter two methods rely on a parallel dataset of melodies and lyrics labeled in Mandarin, which is not available. We identify Cantonese songs from an unlabeled parallel dataset by analyzing the contours between Cantonese tones and melodies, and map the the pairs of absolute melody and tone pitches of lyrics from 6 relative pitches in Cantonese to 3 and 5 relative pitches in Mandarin.

- Fourth, we also present new automatic metrics for lyrics-specific evaluation: intelligibility, contour violation and variation. After defining the metrics, we compare system-created against human-created lyrics in these metrics, including two LLM-based baselines.

We release all code and corpora to the research community.

2. Related Work

2.1. Datasets

Parallel lyrics-melody public datasets are scarce (Ju et al., 2022). One possible solution is to utilize a combination of melody transcription (Yang et al., 2017; Román et al., 2018; Nishikimi et al., 2019), lyrics recognition (Zhang et al., 2022), and techniques for automatic alignment of lyrics to audio (Hosoya et al., 2005; Dabike and Barker, 2019; Suzuki et al., 2019) to reduce the expenses associated with creating large parallel datasets (Watanabe and Goto, 2020). Our method models the relationships between melody and lyrics explicitly, based on knowledge from phonetics and musicology, which the above methods do not.

2.2. Lyrics Generation

Lyrics generation has been explored in a wide range of languages such as English (Manjavacas et al., 2019; Sheng et al., 2021), Japanese (Watanabe et al., 2017, 2018), Greek (Lampridis et al., 2020), and Portuguese (Oliveira et al., 2007; Oliveira, 2015). In the domain of Mandarin lyrics generation, many models follow the sequence-to-sequence (Seq2seq) machine translation model for classical poetry generation (He et al., 2012; Yi et al., 2017). In this method, the generators use the one line of a poem as the source language for the next line, which is regarded as the target language.

As aligned lyrics-melody datasets are hard to come by, methods were developed which require only lyrics to train lyrics generation models. Li et al. (2020) introduced a generator called SongNet with sets of symbols such as format and rhyme, intra-positions, and segment symbols to generate Songci, which is a classical poetry sung in a collection of melody templates in historical China. Their work was extended by Liu et al. (2022) by adding word granularity and reverse order embeddings. The latter method is designed to model rhyme more explicitly. It has also been applied for the generation of Mandarin Rap lyrics (Xue et al., 2021). Giving the systems a set of keywords as input is a common method to "set the theme" of the lyrics. Zhang et al. (2020) expanded this input by allowing passage-level text as "keywords".

Systems using parallel datasets include the system by Lu et al. (2019), who trained a Seq2seq model based on Recurrent Neural Networks (RNN) with 50,000 songs with both lyrics and music notation. They used existing melodies to generate lyrics in the evaluation, which might have a potential influence on subject rating. Another Seq2seq model is iComposer (Lee et al., 2019), which is available online. It is based on Long Short-Term Memory (LSTM) and was trained to generate lyrics and melody bidirectionally, using 1,000 aligned lyrics and songs. The model generates a sequence of pitch and duration of notes from lyrics, but does not use any duration information during generation. Similarly, the model is trained on absolute rather than relative pitch, which can cause problems with some input melodies. During inference, if the input pitch is not located in the model's expected range, the model fails to generate any lyrics. In addition, as the model operates on only one sentence of melody at a time, no thematic connection among adjacent lines can be established.

2.3. Automatic metrics for lyrics generation

Choi (2018) outlined computational approaches such as traditional text metrics like word frequency, familiarity and concreteness to evaluate the complexity and imagery of lyrics. The study however concluded that concreteness is not a good metric as it is always low in certain types of lyrics (e.g., those of love songs and those containing many figurative expressions). Therefore, the method might work better with additional data such as topics and genres of songs. Due to the paucity of human evaluation results on lyrics, Choi (2018) proposed a method called Lyric Topic Diversity Score (LTDS) which utilizes users' interpretations of lyrics to evaluate the complexity of lyrics. The method is based on the assumption that if a lyric shows a large variation of topics, then it is more difficult to understand by listeners.

Li et al. (2020) introduced a set of automatic metrics for the evaluation of generated poetry, which includes aspects of format, rhyme, and sentence integrity. They introduced a "format" metric, which evaluates whether the generated content complies with the specified format, particularly concerning the number of characters in each sentence.

Meanwhile, their "rhyme" metric assesses the presence of rhyme within the generated poetry. By checking for agreement of the last characters in each line of the lyrics, it rewards lyrics that display end rhyme. Sentence integrity is evaluated by fine-tuning a GPT-2 model to predict the probability of punctuation.

3. Method

3.1. Music theory

When producing a new song, one can either begin by composing the lyrics or by creating the melody. If lyrics are to be written for a given melody, the implementation of scansion becomes valuable in examining the compatibility of the lyrics with the melody. This ensures intelligibility and singability throughout the song.

The use of scansion is not limited to English, a non-tonal language. It can be expanded to tonal languages such as Mandarin. Mandarin is characterized by five distinct tones that can distinguish the meaning of Hanzi having the same pinyin[2], as shown in Fig. 2.

Tone	Hanzi (pinyin)	Translation
1	妈(mā)	Mother
2	麻(má)	Numb
3	马(mǎ)	Horse
4	骂(mà)	Scold
0	吗(ma)	What

Figure 2: Five tones in Mandarin

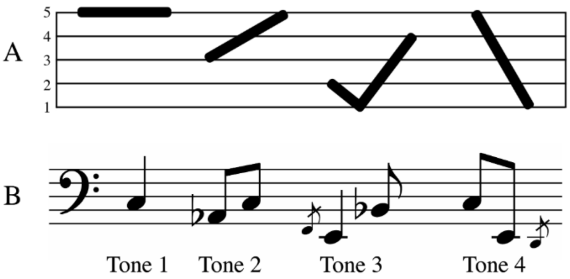

Figure 3: Four tones in Mandarin as 5-level tone contours (A) and Western musical notation (B). The fifth tone (tone 0) has no pitch and is excluded.

Lijia Wang (1993) described Mandarin tones as contours in a 5-level notation relative pitch (Fig.3 A introduced by Chao (1933)) and as music score (Fig.3 B). This way of representing tone can be seen as an expansion of scansion to Mandarin.

Our idea is that scansion can be conducted in both directions, from melody to scansion (M-scansion) and from lyrics to scansion (L-scansion). This way, it constitutes an intermediate representation that connects melody and lyrics. Different representations are possible, depending on whether we represent tones in 3 heights or 5 heights. 3-height scansion classifies tones and melody into 3 levels: high, middle and low. Meanwhile, Mandarin tones are naturally categorized in 5 height, so t_1, t_2, t_3, t_4, t_0 are used in the 5-height scansion.

[2] Pinyin is the romanization system for Standard Mandarin Chinese.

3.2. M-scansion

We investigate four methods of creating M-scansion from melody: heuristic rules based on neumes, contour matching using cosine similarity, sequence labeling through a hidden Markov model, and sequence completion using GPT-2. We empirically test whether 3-height and 5-height scansion is the better representation, by applying these methods to both. Out of our four M-scansion methods, HMM and GPT2 require parallel data for training, whereas neume and similarity metric doesn't need to be trained.

Absolute pitch from the melody need to be mapped to relative pitch in the M-scansion. The translation cannot proceed note for note, but must take the context into account, because the contour of melody is more important than pitch distance between adjacent notes. Most people perceive and understand music through relative pitch. They recognize relationship between notes in a melody (e.g., intervals) regardless of the absolute pitches being played. In addition, notes represented in absolute pitch is sparse because it spans a wide range of possible values (notes) and may have uneven distributions. Converting absolute pitches to relative pitches can make the values denser without affecting the distance between pitches. An example of pitch conversion is illustrated in Tab.4. The conversion is based on the difference between each pitch and the minimum pitch value.

Pitch	74	73	69	74
Converted	5	4	0	5

Figure 4: Conversion from absolute pitch of melody to relative pitch

We do not have a ready-made parallel dataset for Mandarin available. This is because current Mandarin pop songs cannot be used as they do not sufficiently observe the tone-melody matching that we want to create. We turn to Cantonese songs as these obey tone-melody matching. In a way, we learn from Cantonese songs the properties we would like Mandarin songs to have. We determine which songs in the iComposer parallel dataset are likely to be Cantonese. We do this by translating Cantonese tones into contour heights. We consider 3 different representations of Cantonese tones, all of which operate in 5-height, as described in Table 1.

We plot similarity measured by cosine similarity, for each of the three representations in the iComposer dataset and find a bimodal distribution as shown in Figure 5 (blue: 3-tone; red: 5-tone; green: 6-tone). We choose the lyrics with high similarity (≥ 0.6) for 6-tone as our training material[3].

Tone	Pitch	6-tone	5-tone	3-tone
1	5-5	5	5	5
2	3-5	4	4	5
3	3-3	3	3	4
4	2-1	1	1	1
5	2-3	2.5	3	3
6	2-2	2	2	3
7	5	5	5	5
8	3	3	3	3
9	2	2	2	3

Table 1: 6/5/3-tone representation (Cantonese)

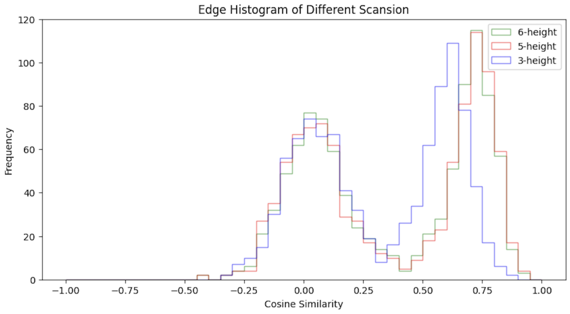

Figure 5: Similarity of pitch and Cantonese lyrics in iComposer dataset, measured by 6/5/3-tone Cantonese scansion

We manually confirm that lyrics with high similarity are indeed Cantonese, by inspecting 20 randomly chosen songs out of the 415. All of these were Cantonese. We also sampled 20 lyrics from the group below 0.6 and found that all were Mandarin or Min Nan dialect.

The 415 songs are preprocessed by filtering out lines that start with t_0 in 5-height because a Mandarin sentence must not begin with a neutral tone. The absolute pitch values in the dataset are shifted so that the minimum pitch in the sequence is 0. The resulting dataset for training M-scansion of HMM and GPT2 has 7,502 lines for training, and 833 lines for testing (8,335 lines in total).

3.2.1. Neume detection

A neume is a fundamental element in the musical notation of chant songs, providing an abstract representation of how a text should be sung in relation to relative pitch (Parrish, 1978). The longest basic shape of a neume contains three notes, as shown in Fig.6.

Apart from the traditional three-note neumes, we add 5 shapes of triplets where three notes with same pitch and another four triplets contain two notes with same pitch. We determine the first element by comparing it to the median pitch of the melody (default median contour segment is $\langle m \rangle$ and $\langle t_2 \rangle$). The pitch gap is assessed in relation to the pitch difference, as depicted in Figure 2. We measure the pitch distance in semitone, representing the gap between Tone 1 and Tone 2, which re-

[3] For 3-tone, we find 255 songs with similarity above 0.6, and 425 for 5-tone.

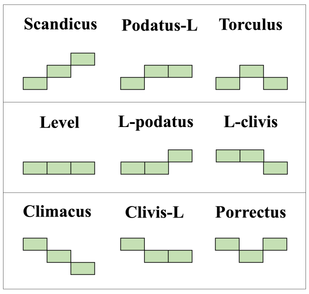

Figure 6: Basic three-note neumes

sults in a pitch difference of (4-3) * 2 = 2 in 5-height scansion. After determining the first element, the group of the first three notes is then mapped to a three-note neume shape. Subsequently, the following two notes in the group are assigned scansion symbols based on the pitch distance. The window size of 3 moves forward to the next set of triplets to identify the neume once again.

3.2.2. Maximum similarity

Mandarin tones consist of 2-3 pitches. In order to calculate the cosine similarity between the pitch sequence of the lyrics and the pitch sequence of the melody, the pitches can be simplified into their primary pitch, as shown in Tab. 2.

Tone	1	2	3	4	0
Primary pitch in 3-height	5	3	1	5	1
Primary pitch in 5-height	4	3	1	5	0

Table 2: Primary relative pitch of tones in Mandarin

The primary pitch of Tone 1 is assigned a value of 4 due to tone sandhi (a phenomenon where tones in speech are systematically changed in certain contexts). In Mandarin, one kind of tone sandhi occurs when two adjacent Hanzi characters are both in Tone 1. The pitch of the first Hanzi in Tone 1 then decreases from 5-5 to 4-5.

There are $4 \times 5^{n-1}$ candidates in 5-height scansion, because the first tone cannot be t_0, and 3^n candidates in 3-height scansion for a melody with n notes. To streamline the search process and reduce computational time, a window size of 6 is used for 5-height similarity, while a window size of 4 is employed for 3-height similarity. The primary objective is to maximize the cosine similarity.

3.2.3. Sequence Labelling by HMM

Our next M-scansion method is to use a Hidden Markov Model (HMM) to predict M-scansion in a sequence labeling task. The model's state space corresponds to the tonal symbols (H symbols for H-height scansion), while the observations represent the pitches in the sequence (26 pitches after shifting from absolute pitch in the dataset).

The model is trained using the Baum-Welch algorithm with labelled data. The Viterbi algorithm is applied to decode the most probable sequence of tonal symbols for a given pitch sequence. Model performance is evaluated through our new automatic metric called contour violation, which we introduce below.

3.2.4. Sequence Completion by GPT2

The last M-scansion method involves finetuning a GPT2-medium model for 6 epochs, using a batch size of 8 and Cross-Entropy Loss with Adam optimization, on a parallel melody-lyrics dataset.

3.2.5. M-Scansion evaluation

We propose a new automatic metric to measure how close a contour of lyrics is to a given melody, called contour violation. The best-possible score of contour violation metric is 0, which means that no violation was detected. The score is calculated line-by-line and nomalised by length (number of Hanzi per line). Examples for three ways of labelling contour tendency symbols are shown in Tab.3, Tab.4 and Tab.5.

Pitch	64	72	72	68	70
Contour segment	<bos>	<rise>	<flat>	<fall>	<rise>

Table 3: Translation from Pitch to Contour

Lyrics	窗	外	的	麻	雀
5-height	t_1	t_4	t_0	t_2	t_4
Contour segment	<bos>	<rise>	<fall>	<rise>	<rise>

Table 4: Translation from Tone to Contour (5-height)

Lyrics	窗	外	的	麻	雀
3-height	h	h	l	m	h
Contour segment	<eos>	<flat>	<fall>	<rise>	<rise>

Table 5: Translation from Tone to Contour (3-height)

If there is a contradiction between corresponding labels in the same position, i.e. <*rise*> vs. <*fall*>, we subtract one from the score. The best possible performance is therefore 0 in this score.

Tab. 6 shows the results. Cosine similarity performs significantly better than other methods in M-scansion, in both 3 and 5-height representation ($p < 0.01$[4]), so we adopt it going forward.

3.3. L-scansion

L-scansion converts tones of Hanzi in lyrics into contour symbols. Fig. 3 above showed the different pitches and contours in Mandarin. We split tones

[4] We use a two-tailed paired MC permutation test with $\alpha = 0.05$ and $R = 10,000$.

	Accuracy		Contour violation	
	3-ht	5-ht	3-ht	5-ht
Neume	0.41	0.29	2.68	2.49
Similarity	0.57	0.35	1.66	0.71
HMM	0.57	0.24	4.45	3.30
GPT2	0.56	0.32	3.77	3.59

Table 6: Automatic evaluation of M-scansion type

into three groups based on their primary pitches: low $<l>$, middle $<m>$ and high $<h>$. This is illustrated in Tab. 7.

Lyrics	妈	麻	马	骂	吗
Tone	1	2	3	4	0
5-height	t_1	t_2	t_3	t_4	t_0
3-height	h	m	l	h	l

Figure 7: L-scansion in 3/5-height

We use a Chinese lyrics corpus of 36,891 lyrics from 494 singers[5]. In this corpus, four lines together form a group. A total of 408,985 lyrics groups are included in the dataset, with a train/dev/test split of 327,133 / 40,288 / 41,564.

4. Overall Model Design

Our overall model, illustrated in Fig. 8, consists of two parts: a melody processor and a lyrics generator. Given a new melody, the melody processor first extracts pitch and duration of each musical note by pretty midi[6]. The pitch number is mapped to a key in a piano keyboard.

We finetune mbart-large-cc25, a multi-lingual sequence-to-sequence model (Liu et al., 2020), to generate Mandarin lyrics. As a variant of the BART (Bidirectional and Auto-Regressive Transformers) (Lewis et al., 2020), mBART is pre-trained on a diverse corpus comprising 25 languages, including Simplified Chinese.

The pseudo-melody, as the source language, is composed of three key elements: a keyword, the Hanzi count, and the scansion sequence (in a 3/5-height ratio), while the lyrics are in the target language (Mandarin).

The model undergoes training with a batch size of 32 over a span of 3 epochs, employing cross-entropy loss as the optimization objective.

5. Evaluation Metrics

Apart from our new metric contour violation, which was introduced in section 3.2.5, we introduce further new metrics for the automatic evaluation of intelligibility and variation of lyrics.

5.1. Intelligibility

Munro and Derwing (1995) defined intelligibility as the extent to which a listener can understand a given speech. As human evaluation is time-consuming and expensive, automatic proxy metrics are attractive; automatic intelligibility scores (Holube and Kollmeier, 1996) have been routinely used in automatic speech recognition (ASR) (Karbasi and Kolossa, 2022). We transfer the method to songs. Lyrics generated by our competitor systems are sung out by a female virtual singer from Synthesizer V Studio[7], using the same melodies that we used for generation. As speech recogniser, we use CapCut[8], a software by TikTok, and report deviation from the original text as an error for each line in the lyrics (reported as % of characters).

5.2. Variation

Variation is important for a lyricist, who should be able to create distinct lyrics that suit different melodies and moods. We measure the variation amongst several lyrics produced by the same system using the cosine similarity of the embeddings generated by text2vec-base-chinese[9].

5.3. Topic fit

We give our system keywords to “set the mood”, as is common in the field of lyrics and poetry generation. Our topic fit metric is designed to measure the degree to which lyrics obey this prompt, and expresses this as a similarity metric (cosine of the embeddings mentioned above). We want to punish systems that do not react to keywords at all, as the ability to request a certain topic is an important control we would like to have over the lyrics. However, when observing humans, we find that while they take the keywords into account very well, they might not repeat the keywords themselves, but find metaphors or paraphrases to express the topic in more subtle ways. The ideal point on the topic fit scale is therefore somewhere in the middle. This was our reason for defining topic fit as the difference from the human level of topic fit.

6. Experiment

We use our new evaluation metrics to evaluate the performance of our scansion-based lyrics generators SmBART-3 and SmBART-5 (trained on 3 or 5-height scansion, respectively) in comparison to four baselines: GPT2-lyrics, iComposer, SongNet , and GPT-3.5 Turbo with prompting.

[5] The Chinese lyrics corpus is available at https://github.com/gaussic/Chinese-Lyric-Corpus

[6] https://github.com/craffel/pretty-midi

[7] https://dreamtonics.com/synthesizerv/

[8] https://www.capcut.com/zh-tw/

[9] https://text2vec.org/

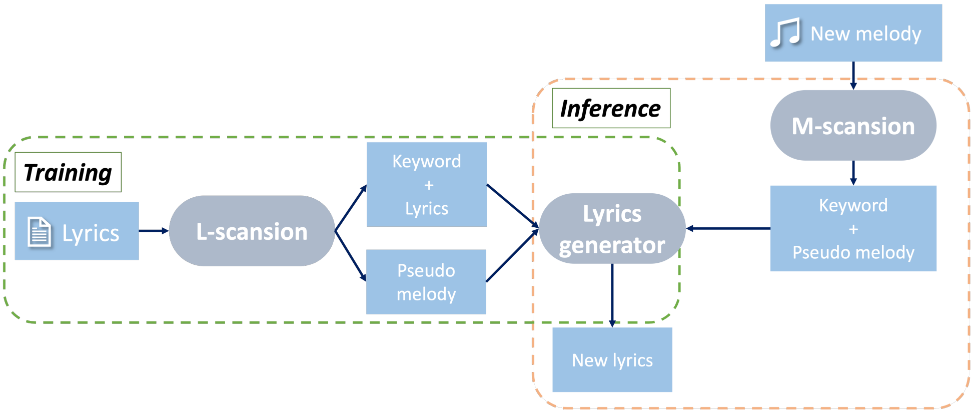

Figure 8: Our scansion-based model

The GPT-2 Chinese lyrics model was finetuned on a pre-trained model (Zhao et al., 2019) by 150,000 Chinese lyrics, using a dataset nearly four times the size of ours. This model uses keywords provided in groups as prompts to generate lyrics. The number of musical notes from the given melodies is a threshold for truncating the generated output, with punctuation being removed from the generated lyrics.

iComposer system takes the pitch of each line within a given melody as an input to generate lyrics in traditional Chinese Hanzi, which are subsequently automatically converted into simplified Chinese Hanzi using OpenCC[10]. In the case of our melodies, the pitch information is extracted and provided to iComposer. It's important to note that iComposer relies solely on pitch information for lyric generation, and therefore, does not require the use of keywords.

SongNet is trained using our dataset. However, SongNet does not accept melodies in MIDI format. To handle the provided melodies, we extract the number of Hanzi in each line of melody and combine them into a sequence of 4 lines by sentence segments used for SongNet. This information is utilized for maintaining the format, ensuring rhyme, and creating segment embeddings within SongNet. Furthermore, in the training process, we employ keywords instead of Cipai[11].

We first use GPT-3.5 Turbo with zero-shot setting to generate Mandarin lyrics per line. The prompt we use is shown in Tab. 8.

We also invite five professional lyricists with over 8 years of experience to participate in the competition. Each lyricist is provided with keywords and melodies to work with. They are each asked to write lyrics for two melodies, each within a one-hour time constraint, which prevents excessive refinement of their lyrical content.

To generate keywords, we apply the TextRank (Mihalcea and Tarau, 2004) algorithm to a dataset of 15,000 Mandarin lyrics crawled from the web. From the returned words, we select the 40 most frequent ones, 4 for each melody, based on their frequency in this dataset, as keywords for our experiment. In cases where multiple keywords have the same English meaning, we remove redundant duplicates. We randomly select 4 keywords for each of our 10 melodies. Table. 7 shows some examples of the sets of keywords we create this way.

黑夜(night), 无情(ruthless), 泪水(tear), 眼神(eye spirit)
风雨(wind and rain), 结果(result), 心情(mood), 朋友(friend)
梦想(dream), 世间(world), 地方(place), 流浪(straying)

Table 7: Example keyword groups

While creating new materials by actual composers incurs additional costs, we chose this approach to enhance the meaningfulness of our evaluation compared to using existing melodies. Utilizing existing melodies carries the risk of introducing bias, as the lyrics corresponding to a given melody might already be included in the training dataset. We prepare 10 new melodies encoded in MIDI. To mitigate the potential influence of the melody, we regulate the vocal range and tempo of each composition[12]. This range covers the lower register of the Contralto and extends to the uppermost notes of the Soprano (Peckham, 2005). The number of note in each line of melody is from 7 to 12.

7. Results

Figure. 9 presents the results of our evaluation using the new metrics. Our systems are designed for maximum tone–melody match, so we report **contour violation** results first. We can see that SmBART-5 (1.83% contour segment error rate) sig-

[10] https://github.com/BYVoid/OpenCC

[11] A title representing a tonal pattern in classical Chinese poetry.

[12] BPM (beats per minute) of melodies was 120. Vocal range stretches from E3 to C6.

System	**Contour Violation**	**Intelligibility**	**Variation**	**Perplexity**	**Topic fit**
Metric	# wrong c-seg (%)	# wrong char (%)	Cosine	bits	Δ Cosine
Better is...	lower	lower	lower	higher	closer to human
GPT2-lyrics	4.85	8.27	.637	.69	−.08
iComposer	4.65	20.91	.718	.68	−.13
SongNet	5.13	3.00	.636	.67	+.07
GPT3.5-P	4.78	11.77	.685	.71	+.08
SmBART-3	**2.65**	2.41	**.628**	.71	+.05
SmBART-5	**1.83**	3.40	.658	.73	+.06
Human	4.83	6.90	.637	.73	+.00

Figure 9: Automatic evaluation results (boldfaced is best automatic system, if significantly different from next-best system)/

nificantly outperforms all other systems. SmBART-3, the next best system at 2.65%, is significantly better than the next best automatic system (iComposer at 4.65%). In fact, GPT2-lyrics, iComposer, SongNet, GPT3.5-P and humans are indistinguishable on this metric. The human "ceiling" does not act as a ceiling here: our expert lyricists' lyrics show a relatively high contour violation of 4.83%. We didn't ask the humans specifically to fit the lyrics to the melody and they were under time pressure when they wrote the lyrics. While songs with good melody–lyric fit are pleasant to listen to, they also require a higher level of effort for humans, as the search space is so large. We therefore do not take the fact that our experts didn't produce such lyrics as an indication that systems should not aim for the fewest possible contour variations. Out of all automatic systems, SongNet had the highest contour violation rate at 5.13% (significantly different from both SmBART models). Both neural systems perform badly, GPT-3.5-P at 4.78% and GPT2-lyrics at 4.85, which is not surprising given their design: these systems do not have access to the melody.

We next turn to **intelligibility**, an important aspect of lyrics quality. In these results, the three numerically best systems – SmBART3 (2.41%); SongNet (3.00%); SmBART-5 (3.40%) – are indistinguishable from each other. However, out of the three, only SmBART-3 is significantly better than GPT2-lyrics, and it is also marginally different from the humans. These pairwise significance results, taken together, show a slight preference for SmBART3. GPT3.5P is significantly worse than SmBART3, SmBART5 and SongNet.

The intelligibility metric does not have a clear winner, but it has a clear loser. iComposer's error rate is more than 1000% that of SmBART-3. It is significantly worse than all systems except from GPT3.5-P. The reason is for iComposer's bad performance may have to do with the fact that it was trained using a melody-lyrics parallel dataset of 1,000 songs without language labels, half of which are Cantonese songs and the other half songs written in Mandarin and other languages. The small size of training dataset prevents iComposer from generating fluent sentences. Also, the vocabulary and syntax between Mandarin and the other languages is different. If iComposer generates lyrics that are Cantonese rather than Mandarin, it gets hurt by the fact that the synthesiser we use is singing in Mandarin pronunciation. We also exploit Cantonese lyrics, but we avoid problems with vocabulary and syntax because we the only information we use from Cantonese are the tones and their fit to the melody.

Let us now turn to our proxy metrics for creativity. In our **variation** metric, Human lyrics show the best performance, as expected. SmBART-3 (.628), SongNet (.637) and GPT2-lyrics (.637) are joint winners (indistinguishable amongst themselves), but only SmBART-3 significantly better than the next-best tier of systems SmBart-5 (.658) and GPT-3.5 (.685). SmBART-5's relatively low performance here might come from the fact that it is limited by the strict matching between tone of lyrics and melody, which reduces the number of the lyric candidates that fits the melody. Running out of candidates is one of the biggest dangers for generative models such as ours. In comparison, SmBART-3 model shows a better balance between melody–tone matching and variation. The lowest performer is iComposer at .718 (significantly different to all other models).

The next creativity metric is **perplexity**. All systems perform in the range of .67-.73, with SmBART-5 and humans jointly at the high range. The only significant difference we can establish in this metric, however, is that iComposer is worse than all other systems. High perplexity can mean that the sentence is creative or that it is not fluent. We conclude that this is maybe the least informative of our creativity metrics.

When it comes to **topic fit**, we can see that SmBART-3, SmBART-5, SongNet and GPT3.5-P are indistinguishably close to the human "medium fit" to keywords, at .05, .06, .07, and .08 respectively. These four systems, however, are significantly better than GPT2-lyrics and iComposer, which is not surprising as these systems cannot take keywords as their input.

Taking all metrics into account, we feel that SmBART-3 has the best overall profile. SongNet and SmBART-5 are also not bad, but while SmBART-3 is the numerically best performer in all metrics except Contour violation, and even in that category it is statistically indistinguishable from the numerically best system (SmBART-5). Large language models, at least without extensive prompting, fared less good in our experiment. We believe this demonstrates advantages of our architecture, which offers direct symbolic control over crucial aspects of lyrics generation. SongNet has different strengths and weaknesses. It can not use melody directly, but it has a sophisticated segment embedding to ensure the integrity of generated lines.

Our prompting experiments with GPT-3.5-P show mixed results. We find that the proper way of prompting made a huge difference, and we designed a way to make it generate lyrics with the correct number of Hanzi characters. However, like GPT-2 lyrics, many of its lines exhibit the same issue of being cut off due to a sentence length threshold.

8. Conclusion

We have shown that our scansion-based hybrid models are able to generate lyrics that are better than competitors in the matching between tonal contour and melody contour (161% and 81% improvement in error rate to next-best system, respectively). This was our main motivation behind the creation of the scansion method. Scansion works by comparing the scansion corresponding to the melody (M-scansion) with a scansion corresponding to the lyrics (L-scansion). We are also interested in a song's potential for dissemination. Lyrics hat are easy to understand are more likely to be remembered and sung, so more intelligible songs are preferable. Our automatic method of measuring intelligibility is based on a synthesizer and a STT model. We also present a set of automatic metrics that can potentially be used to pinpoint creativity, although the results are less conclusive than those for intelligibility and tone–melody match.

One of our contributions concerns corpus-building. We present a method for creating pseudo melodies from lyrics, enabling the creation of potentially very large parallel pseudo-melody–lyrics parallel datasets. These can stand in for melody–lyrics datasets in many supervised situations. One of the tricks we used in gathering information for our M and L-scansion is to use a different tonal language, Cantonese, to estimate scansion matching. The fact that this transfer from Cantonese to Mandarin lyrics works so well indicates the possibility of a universal connection between melody and lyrics. Practically, our use-case is one more demonstration that low resource languages can derive advantages from analogous datasets originating from other languages, in this case one with an abundance of parallel datasets.

Finally, our prompting strategy worked well only when we forced very specific output requirements onto GPT-3.5, namely a tabular format. We hypothesize that this could be effective for not only for languages where the minimum singing unit is a character (as in Mandarin), but also for languages with syllabic structures, such as English.

9. Limitations

Our model is based on the idea of scansion that is only influenced by the pitch of musical note and relative pitch of Hanzi. The model does not use the note duration which is also important information for lyricists during writing. The model may generate lyrics that fit the melody's pitch but fail to match the intended rhythm, which can also cause Mondegreen. It ignored the influence of consonant and vowel of Hanzi as well.

Our model generates lyrics line by line. Four lines, as a chorus, might be logically or thematically inconsistent.

The models used in the task of lyrics generation often assume that Hanzi and musical note are mapped one-to-one. However, in actual songwriting, no matter in tonal and non-tonal languages, a one-to-many mapping between syllables and notes is common. This oversimplification to one-to-one alignment can hinder the model's ability to generate lyrics resembling those of actual songs.

Rhyming is widely regarded as a feature of lyrics, making them easier to remember. Our model cannot guarantee that the generated lyrics will rhyme.

The keywords selected for the experiments were chosen because of their high frequency as keywords in each line. In real songs, core keywords often appear in a title or chorus, rather than in every line of the lyrics.

10. Ethics Statement

The application scansion might influence the protection of the copyright of songwriters. Our lyricists contributed voluntarily and our experiment passed our internal ethics review. We respect the copyright of the lyricists and no personal data is held in any form.

11. Acknowledgements

We thank three anonymous reviewers for their insightful feedback

12. Bibliographical References

Yuanren Chao. 1933. *Tone and Intonation in Chinese*. Bulletin of the Institute of History and Philology.

Kahyun Choi. 2018. Computational lyricology: quantitative approaches to understanding song lyrics and their interpretations.

Gerardo Roa Dabike and Jon Barker. 2019. Automatic lyric transcription from karaoke vocal tracks: Resources and a baseline system. In *Interspeech*, pages 579–583.

Roland Greene and Stephen Cushman. 2016. *The Princeton handbook of poetic terms*. Princeton University Press.

Jing He, Ming Zhou, and Long Jiang. 2012. Generating chinese classical poems with statistical machine translation models. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*.

Inga Holube and Birger Kollmeier. 1996. Speech intelligibility prediction in hearing-impaired listeners based on a psychoacoustically motivated perception model. *The Journal of the Acoustical Society of America*, 100(3):1703–1716.

Toru Hosoya, Motoyuki Suzuki, Akinori Ito, Shozo Makino, Lloyd A Smith, David Bainbridge, and Ian H Witten. 2005. Lyrics recognition from a singing voice based on finite state automaton for music information retrieval. In *ISMIR*, pages 532–535.

Zeqian Ju, Peiling Lu, Xu Tan, Rui Wang, Chen Zhang, Songruoyao Wu, Kejun Zhang, Xiang-Yang Li, Tao Qin, and Tie-Yan Liu. 2022. Telemelody: Lyric-to-melody generation with a template-based two-stage method. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 5426–5437. Association for Computational Linguistics.

Mahdie Karbasi and Dorothea Kolossa. 2022. Asr-based speech intelligibility prediction: A review. *Hearing Research*, page 108606.

Orestis Lampridis, Athanasios Kefalas, and Petros Tzallas. 2020. Greek lyrics generation. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*, pages 445–454. Springer.

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, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Piji Li, Haisong Zhang, Xiaojiang Liu, and Shuming Shi. 2020. Rigid formats controlled text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 742–751, Online. Association for Computational Linguistics.

Huaiqing Fu Zhen Ma Peicheng Su Lijia Wang, Jianming Lu. 1993. *Modern Chinese*. The Commercial Press.

Nayu Liu, Wenjing Han, Guangcan Liu, Da Peng, Ran Zhang, Xiaorui Wang, and Huabin Ruan. 2022. Chipsong: A controllable lyric generation system for chinese popular song. In *Proceedings of the First Workshop on Intelligent and Interactive Writing Assistants (In2Writing 2022)*, pages 85–95.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Xu Lu, Jie Wang, Bojin Zhuang, Shaojun Wang, and Jing Xiao. 2019. A syllable-structured, contextually-based conditionally generation of chinese lyrics. In *Pacific Rim International Conference on Artificial Intelligence*, pages 257–265. Springer.

Enrique Manjavacas, Mike Kestemont, and Folgert Karsdorp. 2019. Generation of hip-hop lyrics with hierarchical modeling and conditional templates. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 301–310.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411.

Murray J Munro and Tracey M Derwing. 1995. Foreign accent, comprehensibility, and intelligibility in the speech of second language learners. *Language learning*, 45(1):73–97.

Ryo Nishikimi, Eita Nakamura, Masataka Goto, and Kazuyoshi Yoshii. 2019. End-to-end melody note transcription based on a beat-synchronous attention mechanism. In *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 26–30. IEEE.

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.

Carl Parrish. 1978. *The notation of medieval music*, volume 1. Pendragon Press.

Anne Peckham. 2005. *Vocal workouts for the contemporary singer*. Hal Leonard Corporation.

Miguel A Román, Antonio Pertusa, and Jorge Calvo-Zaragoza. 2018. An end-to-end framework for audio-to-score music transcription on monophonic excerpts. In *ISMIR*, pages 34–41.

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 *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 13798–13805. AAAI Press.

Motoyuki Suzuki, Sho Tomita, and Tomoki Morita. 2019. Lyrics recognition from singing voice focused on correspondence between voice and notes. In *INTERSPEECH*, pages 3238–3241.

Kento Watanabe and Masataka Goto. 2020. Lyrics information processing: Analysis, generation, and applications. In *Proceedings of the 1st Workshop on NLP for Music and Audio (NLP4MusA)*, pages 6–12.

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.

Kento Watanabe, Yuichiroh Matsubayashi, Kentaro Inui, Tomoyasu Nakano, Satoru Fukayama, and Masataka Goto. 2017. Lyrisys: An interactive support system for writing lyrics based on topic transition. In *Proceedings of the 22nd international conference on intelligent user interfaces*, pages 559–563.

Lian Hee Wee. 2007. Unraveling the relation between mandarin tones and musical melody. *Journal of Chinese Linguistics*, 35(1):128.

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. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 69–81. Association for Computational Linguistics.

Luwei Yang, Akira Maezawa, Jordan BL Smith, and Elaine Chew. 2017. Probabilistic transcription of sung melody using a pitch dynamic model. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 301–305. IEEE.

Xiaoyuan Yi, Ruoyu Li, and Maosong Sun. 2017. Generating chinese classical poems with rnn encoder-decoder. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, pages 211–223. Springer.

Jingbo You and Gang Li. 2015. *The Brief History of Chinese Pop Music*. Shanghai Music Publishing House.

Jin'en Yu. 2007. *Republic of the phonetic alphabet policy history theory*. Zhonghua Book Company.

Chen Zhang, Jiaxing Yu, LuChin Chang, Xu Tan, Jiawei Chen, Tao Qin, and Kejun Zhang. 2022. Pdaugment: Data augmentation by pitch and duration adjustments for automatic lyrics transcription. In *Proceedings of the 23rd International Society for Music Information Retrieval Conference, ISMIR 2022, Bengaluru, India, December 4-8, 2022*, pages 454–461.

Rongsheng Zhang, Xiaoxi Mao, Le Li, Lin Jiang, Lin Chen, Zhiwei Hu, Yadong Xi, Changjie Fan, and Minlie Huang. 2020. Youling: an ai-assisted lyrics creation system. In *Proceedings of the*

2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 85–91.

Zhe Zhao, Hui Chen, Jinbin Zhang, Xin Zhao, Tao Liu, Wei Lu, Xi Chen, Haotang Deng, Qi Ju, and Xiaoyong Du. 2019. Uer: An open-source toolkit for pre-training models. *EMNLP-IJCNLP 2019*, page 241.

A. GPT3.5 prompt

Act as a professional Mandarin lyricist.
The requirements of the format how you write the lyrics:
1. Generate a line of lyrics without punctuation.
2. The line should be generated in a table labelled from 1 to N.
3. Each cell contains only a single Hanzi.
4. The keyword for the lyrics is <keyword>.

Table 8: Our prompt to GPT-3.5, using a tabular format of lyrics, with N being the sentence length.

B. Scansion as intermediate for SmBART-3 model

LYRICS	Hanzi	你	们	太	想	开	心
	Tone	T2	T0	T4	T3	T1	T1
	L-scansion	m	l	h	l	h	h
	M-scansion	m	l	h	l	h	h
MELODY	Absolute Pitch	72	68	78	70	76	76

Table 9: Representation of a song. Blue lines are raw input; black lines are created by our scansion analysis in 3-height.