Translate the Beauty in Songs: Jointly Learning to Align Melody and Translate Lyrics

Chengxi Li\textsuperscript{1}, Kai Fan\textsuperscript{2}, Jiajun Bu\textsuperscript{1}, Boxing Chen\textsuperscript{2}, Zhongqiang Huang\textsuperscript{2}, Zhi Yu\textsuperscript{1†}

\textsuperscript{1}Zhejiang Provincial Key Laboratory of Service Robot, Zhejiang University
\textsuperscript{2}Alibaba DAMO Academy

\{chengxili, yuzhirenzhe@zju.edu.cn, bjj@cs.zju.edu.cn\}
\{k.fan,boxing.cbx,z.huang\}@alibaba-inc.com

Abstract

Song translation requires both translation of lyrics and alignment of music notes so that the resulting verse can be sung to the accompanying melody, which is a challenging problem that has attracted some interests in different aspects of the translation process. In this paper, we propose Lyrics-Melody Translation with Adaptive Grouping (LTag), a holistic solution to automatic song translation by jointly modeling lyrics translation and lyrics-melody alignment. It is a novel encoder-decoder framework that can simultaneously translate the source lyrics and determine the number of aligned notes at each decoding step through an adaptive note grouping module. To address data scarcity, we commissioned a small amount of training data annotated specifically for this task and used large amounts of augmented data through back-translation. Experiments conducted on an English-Chinese song translation data set show the effectiveness of our model in both automatic and human evaluations.\textsuperscript{1}

1 Introduction

Song translation is a meaningful human endeavor to climb high Tower of Babel for inter-culture exchange. Yet it has not received much attention in the natural language processing (NLP) community despite the advancement of machine translation technologies, especially Neural Machine Translation (NMT) (Bahdanau et al., 2015; Vaswani et al., 2017; Hassan et al., 2018), and the expanding interests of solving real-world problems using artificial intelligence techniques. Challenges include the lack of efficient means to collect parallel lyrics and alignment data, the difficulty of modeling the complex interaction between texts and melody and imperceptive evaluation of scores. While closely related to text translation, song translation is a more involved task. In addition to the general considerations of word choice and word order in translation, human translators of songs need to have a mastery of cultural traditions and the poetic usage of both source and target languages. Furthermore, the translated lyrics need to be properly aligned with the melody, as shown in Figure 1, to maintain the intact beauty of the song, a factor that is indispensable in song translation (Franzon, 2015).

Researchers have explored Singing Voice Synthesis (SVS) (Liu et al., 2022a,c,b) to automate the vocal singing of songs given the input lyrics and scores, which laid the foundation of convenient and perceptive evaluation and a prospective empirical usage of automatically generated songs. However, there is very few previous studies in the direction of Automatic Song Translation (AST). The sole work (Guo et al., 2022) we are aware of focuses on matching tones and rhythms for the translated target words for tonal languages, by imposing constraints during NMT inference. Their direct use of text translation models and strict mapping between notes and tokens, however, is unable to capture the more involved nature of song translation. While the number of notes provides an easy upper bound on the length of translation, the delicate
alignment between lyrics and melody, as observed in Haapaniemi and Laakkonen (2019), should not be dictated solely by simple rigid rules.

In this paper, we propose Lyrics-Melody Translation with Adaptive Grouping (LTAG), the first comprehensive solution to the AST problem, by jointly modeling of both lyrics translation and lyrics-melody alignment within the transformer-based encoder-decoder framework. LTAG incorporates both lyrics and melody in an end-to-end manner and employs an adaptive grouping module to explicitly model the alignment between lyrics and melody. To facilitate training, we produce the first (Chinese-English) bilingual lyrics-melody alignment data set. To address the data scarcity problem, we also generate a large amount of bilingual lyrics-melody data through back-translation of monolingual lyrics-melody alignment data, which is used together with the high quality manual annotations through a curriculum training strategy. Our experiments show that songs translated by LTAG are both faithful to the original lyrics and singable to the melody, as measured by both automatic metrics and human judges majoring in music. Main contributions of this work are as follows:

1. We propose the first joint lyrics translation and lyrics-melody alignment framework LTAG to solve the AST task in a comprehensive manner.

2. We design an adaptive grouping method for monotonic lyrics-melody alignment prediction that helps achieve high-quality lyrics translation and to provide flexible and reasonable lyrics-to-melody alignments in the same auto-regressive process.

3. We produce the first bilingual lyrics-melody alignment data set that will be released publicly to facilitate further research in this field. We also leverage the back-translation and the curriculum learning strategy to boost performance.

4. Our experiments show that LTAG outperforms baselines by a notable margin. Human evaluations indicate that our proposed flexible alignments together with lyrics translation achieves satisfying song translation results.

2 Related Works

Lyrics and Song translations have recently drawn attention from the NLP community. Automatic lyrics translation has been approached by rule-based methods, statistical machine translation methods, finite-state methods with rhythmic and lexical constraints (Gervás, 2002; Manurung, 2004; He et al., 2012), and more recently by neural methods (Ghazvininejad et al., 2016, 2017, 2018). Traditional song translation research has made progress in terms of lyrics translation and lyrics-melody alignment through linguistic knowledge (Haapaniemi and Laakkonen, 2019; Low, 2003, 2008, 2022; Franzon, 2015; Desblache, 2018). Often, the object of these research was artificial songs. These methods pursue lyrics-melody alignment and lyrics translation in separate tracks. Guo et al. (2022) pursue song translation as a type of Constrained Text Translation. Previous works (Hokamp and Liu, 2017; Lakew et al., 2019; Li et al., 2020; Zou et al., 2021) on constraining the decoding process are shown to be effective in performance and convenient in implementation. Others that impose constraints during training, such as adding format embedding (Li et al., 2020), introducing special tags and rescoring length control (Lakew et al., 2019; Saboo and Baumann, 2019), are data-driven methods and show good performance. In this paper, we propose the lyrics translation model with lyrics-melody alignments for domain shift and length control, and overcome the problem of domain mismatch and data sparsity by using monolingual data.

Lyrics Generation with Alignment Prediction, one of the most important tasks in automatic song production, has received much attention recently. In recent works of machine translation, graph neural networks (GNNs) are widely used for alignment prediction between the source and target sentences (Li et al., 2022, 2023). In our researched domain, most of the current works (Lee et al., 2019; Chen and Lerch, 2020; Sheng et al., 2021; Ju et al., 2021; Ma et al., 2021; Xue et al., 2021) adopt the sequence generation method, but with different objectives. Some constrain rhythmic alignment, others theme and target genre. Other works (Sheng et al., 2021; Ju et al., 2021) apply the attention mechanism and find the lyrics-melody alignment via dynamic programming on the attention weights matrix. This method sometimes results in non-monotonic output. Most importantly, it seems that their alignment component is akin to a post-processing module rather than an integrated unit that learns the dynamic alignment such that the lyrics generation is constrained. In our proposal, we take advantage of the monotonic nature of the lyrics-melody alignments and design a light neural network for alignment prediction in parallel to the translation process.
3 Methodology

In this section, we first describe the LTAG as shown in Figure 2. Then, we detail the adaptive grouping method for alignment prediction and explain how we adapt back-translation for the AST task.

3.1 Overall Architecture

We design an auto-regressive translation architecture that jointly performs lyrics translation and lyrics-melody alignment prediction. As shown in Figure 2, it consists of a transformer-based encoder-decoder pack for lyrics translation, two note-pooling embedding layers that embed and do pooling for notes and alignments, and an alignment decoder. The transformer encoder-decoder is pre-trained with a denoising auto-encoder (Lewis et al., 2020) and for the translation task as in Guo et al. (2022). During pre-training, two prefix tokens indicating the translation direction and the text domain are prepended to the source input. The note-pooling embedding layers shown in Figure 3(a) is a module that processes the melody information. The alignment decoder shown in Figure 3(b) is based on our adaptive notes grouping method that dynamically predicts the number of notes to align to a token during auto-regressive decoding.

3.2 Note-Pooling Embedding

The note-pooling embedding layer takes the notes and alignments as input, and outputs the pooled note embedding and alignment embedding. The input note sequence consists of MIDI pitch and duration of each note. The MIDI pitch and duration can be represented as embedding $e_{midi}$ and $e_{dur}$ respectively. We define the $i$-th note embedding:

$$e_{note}^i = e_{midi}^i + e_{dur}^i + e_p^i$$  \hspace{1cm} (1)

where $e_p^i$ is the positional embedding.

We apply non-overlapping mean-pooling on the note embedding sequence according to the alignment information. Specifically, the embeddings of the consecutive notes that align to the same token are averaged. Mathematically, the alignment information $A$ is represented as a binary matrix $M \in \{0, 1\}^{L \times N}$, where $L$ and $N$ denote the sequence length of the tokens and notes. $M_{ji} = 1$ if the $i$-th note is aligned to the $j$-th token. We use $M$ to efficiently calculate the non-overlapping mean-pooling via matrix multiplication, denoted the result as melody embedding $e_{md}$.

$$e_{md} = \text{Non-Overlap-Mean-Pool}(e_{note}, M)$$ \hspace{1cm} (2)

The kernel size of this operation is not fixed but varies according to the row sum of $M$. The detailed calculation can refer to Appendix A.

Because lyrics-melody alignments are monotonic, we encode the alignment more succinctly by calculating the cumulative sum of the number of aligned notes:

$$s = \text{CumSum}(\text{RowSum}(M))$$ \hspace{1cm} (3)

where $s$ is a vector of length $L$. $s_i/N$ then represents the alignment ratio for each aligned note. We next quantize the cumulative alignment ratios by grouping them into equal-size bins over the range $[0, 1]$, and introduce a set of embedding vectors...
finally, the alignment embedding is calculated as follows.

\[ e_{\text{align}}^j = f(E_{\text{ratio}}(s^j/N)) \quad (4) \]

where \( f(\cdot) \) is a simple non-linear layer of causal 1D convolution with ReLU activation, and the number of bins is a hyper-parameter. The motivation is to implicitly constrain the translation by the number of aligned notes. 

The melody embedding and alignment embedding are summed and then added to the original transformer encoder or decoder input.

\[ e_{\text{enc(dec)}} = e_{\text{token}} + e_p + (e_{\text{md}} + e_{\text{align}}) \quad (5) \]

As calculated in Eq. (2), each melody embedding corresponds to one token. In addition, the causal convolution implies that the alignment embedding tensors also have the same length as the text tokens and guarantees each alignment embedding only observes previous ratio embeddings in an auto-regressive manner. It means that on the decoder, this layer can fit perfectly in the teacher-forcing training.

### 3.3 Alignment Decoder

Inspired by the Adaptive Computation Time (ACT) (Graves, 2016), we propose the adaptive grouping module to model lyrics-melody alignment. As shown in Figure 3(b) and 3(c), this module predicts how many consecutive notes should be assigned to the current token.

For \( 1 \leq j \leq L_Y \), let \( y_j \) be the \( j \)-th target token and \( h_j \) be the corresponding hidden state of the last transformer decoder layer. Suppose previous tokens \( y_{j-1:0} \) have been aligned to the first \( n - 1 \) notes, we define the following adaptive grouping process by iterating over index \( k \) (starting from 1) to derive the number of notes aligned to \( y_j \).

\[ S_{re}^j = N - s_{tgt}^{j-1} \]

\[ h_k^j = h_j \]

\[ h_{k+1}^j = g(h_{k-1}^j, e_{\text{align}}(X), h_{\text{align}}(y_{j-1:0}), S_{re}^j, k-1) \]

\[ \alpha_k^j = \sigma(\text{Linear}(h_k^j)) \]

where \( e_{\text{align}}(X) \) and \( e_{\text{align}}(y_{j-1:0}) \) are the alignment embeddings of the full source input and the partial target input respectively, and \( s_{tgt}^{j-1} \) is \( j \)-th element of vector \( s \) in Eq. (3).

We first calculate the residual number of unaligned notes at the current decoding step \( j \) as \( S_{re}^j \). \( e_{\text{align}}(X) \) is fed into an average pooling layer to obtain a single vector, making it always possible to be additive with \( e_{\text{align}}(y_{j-1:0}) \) of variable length. For all the inputs, we apply a multi-layer network \( g(\cdot) \) shown in green in Figure 3(b). Eventually, the sigmoid function \( \sigma(\cdot) \) outputs the halting probability \( \alpha_k^j \) of the intermediate step. The summation of these probabilities represent the likelihood that the current \( k \) notes are aligned to the target token \( y_j \).

Given a hyper-parameter \( \epsilon \) as a small float number (e.g., 0.01), if \( \sum_k \alpha_k^j < 1 - \epsilon \), the adaptive grouping process will continue and re-calculate by incrementing \( k \) and decrementing \( S_{re}^j \). Otherwise, the aligning process halts, and the alignment decoder outputs the number of aligned notes \( K(j) \).

\[ K(j) = \arg\min_K \left\{ \sum_{k=1}^K \alpha_k^j \geq 1 - \epsilon \right\} \quad (6) \]

A positive \( \epsilon > 0 \) guarantees that \( K(j) \geq 1 \), i.e., at least one note is aligned. To define the halting
probabilities of $K(j)$ aligned notes, we introduce the remainder $R(j) = 1 - \sum_{k=1}^{K(j)-1} \alpha_k^j$. In this way, $\alpha_k^j$ and $R(j)$ can be valid probability distributions. Figure 3(c) is an example of how the adaptive grouping works.

In the labeled alignment data, the ground truth of the number of aligned notes for each target token is available, denoted as $\Delta_j$. Instead of minimizing the ponder cost $\sum_j K(j) + R(j)$ as in ACT (Graves, 2016), we optimize the following adaptive grouping loss $L_G$, which could naturally upper bound the token-wise ponder cost via $\Delta_j$.

$$L_G = \left| \sum_j K(j) - N \right| + \sum_j |K(j) - \Delta_j|$$
$$\approx \left| \sum_j (K(j) - (1 - R(j))) - N \right| + \sum_j |K(j) - (1 - R(j)) - \Delta_j|$$

The variable $K(j)$ is discontinuous with respect to the halting probabilities, so we use $1 - R(j)$ in the approximation to make the loss differentiable (more analysis in Appendix B). Additionally, because tokens aligned to more than one notes are infrequent, we add upweighting to the alignment loss of such tokens for model calibration.

$$L_G = \left| \sum_j (K(j) - 1 + R(j)) - N \right| + \sum_j |(K(j) - 1 + R(j) - \Delta_j) \cdot w_j|$$

where $w_j = 1$ if $\Delta_j = 1$ and $w_j > 1$ is a hyper-parameter if $\Delta_j > 1$.

### 3.4 Back Translation with Alignments

Although a data set of a few thousand verses with human translation and annotated with alignment information is useful, its quantity is limited. We therefore adopt the widely used back-translation method (Sennrich et al., 2016) to generate more training data. We crawl the web for more available monolingual song data with alignments and build another pre-trained lyrics translation model with length control that is used to back translate the monolingual data into the source language. The length control ensures that the number of tokens is the same as the number of notes after which a one-to-one source-side alignment can be generated. This way, we obtain a comparatively larger data set with noise on the source side but still accurate information on the target side.

Because the back-translated data are much larger than the human annotated one, we in practice design our data loader by following a curriculum learning way. Initially, the augmented data from back-translation will be mixed with up-sampled the real data from human annotation. In each training epoch, we gradually down-sample the augmented data to raise the ratio of annotated data in the batch. A visualization of the data sampling scheduler is in Figure 6 (See Appendix C).

### 3.5 Training and Inference

After the pre-training stage, we will optimize the whole model by jointly minimizing the loss from the task of lyrics translation and the task of alignment prediction. Note that the SVS model is pre-trained and only used for evaluation. The overall loss is thus:

$$L_{\text{joint}} = \sum_{j=0}^{L_y} \log P(y_j|y_{j-1}, X) + \beta \cdot L_G$$

where $\beta$ is a hyper-parameter to balance the importance between the two tasks. The inference follows the standard beam search for auto-regressive decoding, while only the last generated token and its corresponding notes should be specially taken care of. Details can be found in Appendix D.

### 4 Experiments

In this section, we describe the experiment setup, results and analysis on Chinese↔English song translation.

#### 4.1 Experimental Settings

**Data Sets**

Since there is no publicly available data set with high quality parallel lyrics translation and lyrics-melody alignments, we collect and annotate a data set PopCV (Pop songs with Cover Version) containing both Chinese songs with their English cover version and English ones with Chinese cover version. Since there are no industry standards or published precedence in annotating such a data set, we design an annotation procedure which is time-saving and easy for annotators to carry out. First, we collect the score sheet files of songs from score websites\(^2\). Then the annotators add lyrics to notes according to how songs are sung in the original and its cover version as conventions\(^3\) suggest. We then export the annotated music score files in .musicxml format.

---

\(^2\)https://www.musescore.org and https://www.midishow.com

\(^3\)https://lilypond.org and https://musescore.org/howto
we follow Sheng et al. (2021) and show annotators the resulting score of the song with translated lyrics. To verify the singability in the end-to-end manner, we additionally use an open-source Chinese singing voice synthesis (SVS) model (Liu et al., 2022a) to supply the annotators with an actual audio rendition of the songs for more intuitive feeling.

Evaluation Metrics

The most convincing evaluation of how our model works is whether the translated songs can be sung, understood, and, most importantly, enjoyed. Thus, we follow Sheng et al. (2021) and show annotators the resulting score of the song with translated lyrics. To verify the singability in the end-to-end manner, we additionally use an open-source Chinese singing voice synthesis (SVS) model (Liu et al., 2022a) to supply the annotators with an actual audio rendition of the songs for more intuitive feeling.

We randomly select 20 verses from the test set and show the music sheets and synthesized singing voice (see Appendix E) of each translated verse to five annotators. For automatic evaluations, we use sacreBLEU

<table>
<thead>
<tr>
<th>Models</th>
<th>MOS-T</th>
<th>MOS-S</th>
<th>MOS-Q</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>En→Zh</td>
<td>Zh→En</td>
<td>En→Zh</td>
</tr>
<tr>
<td>Human Ref.</td>
<td>3.83 ±0.06</td>
<td>4.11 ±0.05</td>
<td>3.92 ±0.07</td>
</tr>
<tr>
<td>GagaST</td>
<td>3.66 ±0.06</td>
<td>3.72 ±0.05</td>
<td>3.49 ±0.10</td>
</tr>
<tr>
<td>LTAG-cls</td>
<td>3.66 ±0.05</td>
<td>3.79 ±0.05</td>
<td>3.58 ±0.07</td>
</tr>
<tr>
<td>only bt</td>
<td>3.69 ±0.05</td>
<td>3.80 ±0.04</td>
<td>3.53 ±0.09</td>
</tr>
<tr>
<td>w/o bt</td>
<td>3.64 ±0.05</td>
<td>3.30 ±0.05</td>
<td>2.16 ±0.05</td>
</tr>
<tr>
<td>LTAG</td>
<td>3.71 ±0.05</td>
<td>3.85 ±0.05</td>
<td>3.68 ±0.05</td>
</tr>
<tr>
<td>only bt</td>
<td>3.71 ±0.05</td>
<td>3.80 ±0.05</td>
<td>3.58 ±0.07</td>
</tr>
<tr>
<td>w/o bt</td>
<td>3.69 ±0.05</td>
<td>3.28 ±0.04</td>
<td>3.63 ±0.07</td>
</tr>
</tbody>
</table>

Table 1: The Mean Opinion Score in translation intelligibility and naturalness (MOS-T), singability (MOS-S) and overall quality (MOS-Q) with 95% confidence intervals. The translation direction with † means that audio samples of the translated song for evaluation are generated with the voice synthesis model that is not trained for that target language. So those results are presented in Appendix F and for reference only.

<table>
<thead>
<tr>
<th>Lang</th>
<th>Songs</th>
<th>Lyrics</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>LMD</td>
<td>En</td>
<td>152,991</td>
<td>Back Translation</td>
</tr>
<tr>
<td>Changba</td>
<td>Zh</td>
<td>542,034</td>
<td>Back Translation</td>
</tr>
<tr>
<td>PopCV</td>
<td>En,Zh</td>
<td>79,295</td>
<td>Anotation</td>
</tr>
<tr>
<td>testset</td>
<td>En,Zh</td>
<td>25,629</td>
<td>Anotation</td>
</tr>
</tbody>
</table>

Table 2: Statistics of datasets in our experiments and automatically extract lyrics and their aligned notes. Please refer to Appendix E for details.

For the data used in back-translation, we use LMD (Yu et al., 2021) for English songs with alignments to melody and a data set crawled from Changba App for Chinese songs. We first pre-train two lyrics translation models with length control, one in each direction, and then translate the above two data sets. The translated lyrics are one-on-one aligned to the notes. Two sets of back-translated data are used for training only while testing is done on real data with human annotations. An overview of the data is in Table 2.

Model Configurations

The token embeddings of the Transformer encoder and decoder have dimension of 256 and are shared. In the note-pooling embedding layer, the size of the lookup table for MIDI pitch and duration type is set to 128 and 31. The halting hyper-parameter epsilon in the adaptive grouping process is 0.05. \( w_j \) is 5 when \( \Delta_j > 1 \), and \( \beta \) is 0.8 in the joint loss \( L_{joint} \). The beam size during decoding is 5.

The LTAG model is pre-trained on the WMT data and the crawled lyrics data, including the par-
4.2 Main Results

We compare LTAG with two baseline systems. One is the GagaST system (Guo et al., 2022), which focuses on the tonal aspect of Chinese. The other one is a variation of our model. This variation uses a transformer-layer based classifier (LTAG-cls) instead of our alignment decoder to predict the number of aligned notes. The maximum number of aligned notes is 30, the same as allowed maximum $K(j)$ in alignment decoder. Besides, we show results from the human reference.

4.2.1 Translation Evaluation

We first report the human evaluation metrics (MOS-T) on both Chinese-to-English (Zh→En) and English-to-Chinese (En→Zh) song translation tasks in Table 1. LTAG generally gains improvements among all systems while the gap between different systems and settings is not obvious. It’s partly because the lyrics translation by professionals is usually free translation rather than literal translation. A missing word in different slices can cause negative, neutral or even positive effect. Only obvious semantic deviations or grammatical mistakes lead to certain score decrease. As discussed in MOS-T, automatic metric BLEU may not be a good criterion to compare the machine translation and free translation for lyrics. But we still present the BLEU results in Table 3. We can see that our proposed system LTAG significantly outperforms the recent baseline GagaST by a large margin on both translation directions. As to the variant model LTAG-cls, the LTAG is still slightly better.

4.2.2 Lyrics-Melody Alignment Evaluation

As for lyrics-melody alignment quality, we report the human evaluation metrics (MOS-S) on en-zh translation direction. In Table 1, LTAG considerably outperforms other systems, especially better than the GagaST with simple length control decoding. Notably, the variant version LTAG cls performs worse than other systems, which indicates that more flexible alignments between lyrics and melody bring listening enjoyment to audience when it’s reasonable enough. Otherwise, the flexibility may be counterproductive. We also evaluate the alignment quality by using the histograms of the number of aligned notes in Figure 4. In Table 3, we calculate the Alignment Score between the histograms of each system and the true histograms. The histograms show that the distribution of align-
ments generated by LTAG resemble those of the true alignments while “GagaST” lacks variety by providing only one-on-one alignments between the lyrics and melody. In conclusion, both results demonstrate that the adaptive grouping method shows significant advantage over the length control or simple classifier in predicting reasonable alignment between the translated lyrics and melody.

In Table 1, MOS-Q mainly reflects the overall intelligibility, naturalness, singability and beauty of the song translation. Since the translation and alignment quality both contribute to the final result, the difference between methods seem less visible. But considering the 95% confidence, we can conclude that the LTAG still ranks best.

4.3 Ablation Study and Analysis

We first conduct ablation experiments to study the effects of back translation data with various settings. In Table 1, we have the following findings for LTAG and LTAG-cls. (1) Since the back-translation data is obviously larger than the real annotation data, there is almost no difference if only back-translation data is used for training. This enables the possibility of training our model in unsupervised way. (2) If only the limited supervised data is used, the performance apparently becomes worse. (3) LTAG is consistently better than LTAG-cls in all ablation experiments. In addition, we verify the importance of the novel alignment embedding \(e_{\text{align}}\) by removing it from the note-pooling embedding layer and alignment decoder, and observe a non-negligible decrease on both BLEU and AS.

Some case studies in Figure 5 suggest that, when the tokens in lyrics fall into one-to-many alignments, GagaST usually provides inappropriate lyrics translation or even decodes non-vocal tokens such as comma to meet the length constraint. It will hurt both the translation quality and the singability of the translated lyrics. In contrast, the simple classifier following transformer layers is enough for flexible alignments. However, our evaluation results indicate our light weighted alignment decoder is capable of providing delicate alignments between tokens and notes.

5 Conclusion

In this work, we propose LTAG, a lyrics translation model with lyrics-melody alignments that allows simultaneous generation of target text and alignment to the music notes. We propose an adaptive grouping method that fits in the auto-regressive
translation process. To better train and evaluate our model, we also annotate a new song translation data set PopCV containing English and Chinese songs with their cover version in both languages and with lyrics-melody alignments. For training, we also employ back-translation to leverage the more abundantly available monolingual lyrics data with lyrics-melody alignments in a curriculum learning way. Evaluations with both the automatic and human metrics show that LTAG is capable of producing natural, singable and enjoyable translation results.

Concerns of the Ethical Impacts

This work develops a possible automatic method for song translation. Therefore, if we release our repository and data set, there is the potential of abusing to synthesize score sheets and texts, especially may cause copyright issues. Thus, we choose the dataset license: CC by-nc-sa 4.0. In this paper, we thoroughly discuss strengths and shortcomings of our proposed model and perform a series of experiments to support them. Codes, model checkpoints and data set will be released upon acceptance after desensitization and compliance examination.

Limitations and Future Work

Our work mainly focuses on Chinese and English songs while languages from other language families are not involved. They may lack lyrics and song data to perform data augmentations or to train a SVS system used in our human-evaluation process. Future endeavors may lie in mining and utilizing song translation data from richer sources and more languages.

Acknowledgments

We would like to thank all the anonymous reviewers for the insightful and helpful comments.

This work is supported by the National Natural Science Foundation of China (Grant No. 61972349), the National Key R&D Program of China (No. 2018YFC2002603), Alibaba-Zhejiang University Joint Institute of Frontier Technologies, and Alibaba Group through Alibaba Research Intern Program.

References


Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all...


A Pooling Matrix in the Note-pooling Embedding Layer

We have note embedding $e_{\text{note}} \in \mathbb{R}^{N \times d}$ ($d$ is the embedding dimension) and alignment matrix $M \in \{0,1\}^{L \times N}$. The non-overlapped mean-pooling can be calculated as follows.

$$W = M / \text{sum}(M, \text{dim} = -1, \text{keepdim} = \text{True}) e_{\text{md}} = W * e_{\text{note}}$$

where / is element-wise division and $*$ is matrix multiplication. By leveraging gather and scatter operations, the non-overlapped Mean-pooling can even be computed in batch.

B Analysis of Adaptive Grouping Loss

By the definition of the adaptive grouping loss, we only need to analyze the following term.

$$|K(j) - (1 - R(j)) - \Delta_j|$$

If $K(j) > \Delta_j$ in the forward pass, we have $K(j) - \Delta_j \geq 1$ because they are both positive integers. In order to encourage the loss to become smaller, $1 - R(j) = \sum_{k=1}^{K(j)-1} \alpha_j^k$ should become larger. In other words, the optimization will push $\sum_{k=1}^{K(j)-1} \alpha_j^k$ to be larger towards $K(j) - \Delta_j$. Note that the theoretical upper bound of $\sum_{k=1}^{K(j)-1} \alpha_j^k$ is $K(j) - 1$, which is larger or equal to $K(j) - \Delta_j$. Thus, this optimization is possible and it will meet the following condition during optimization.

$$\sum_{k=1}^{K(j)-1} \alpha_j^k \geq 1 - \epsilon.$$ 

By definition of $K(j)$, we have the following conclusion.

$$K(j)^{\text{new}} = \arg \min_k \left\{ \sum_{k=1}^{K(j)-1} \alpha_j^k \geq 1 - \epsilon \right\} \leq K(j) - 1$$

If $K(j) < \Delta_j$, a similar analysis can be derived. $\sum_{k=1}^{K(j)-1} \alpha_j^k \rightarrow 0$ should be encouraged to push a smaller loss. It implies if the $K(j)$-th halting probability doesn’t satisfy the condition $\alpha_j^{K(j)} \geq 1 - \epsilon$, the $K(j)^{\text{new}}$ will have an increasing trend. However, if $\alpha_j^{K(j)} \geq 1 - \epsilon$, the optimization will be stuck. We may optimize $\left| K(j) - (1 - R(j) + \alpha_j^{K(j)}) - \Delta_j \right|$, i.e.,

$$\left| K(j) - \sum_{k=1}^{K(j)} \alpha_j^k - \Delta_j \right|.$$ 

In practice, we found this is a rare case and it will completely disappear after several epochs. So we adopt the unified adaptive grouping loss.

If $K(j) = \Delta_j$, it means we can safely remove this term in the loss.

C Scheduler of Curriculum Learning

The down sampling ratio of back translation data starts at 1.00 and decrease to 0.01 at the half of total training epochs. The sampling ratio of annotation data starts at 20.00 for upsampling and decrease to 5.00 at the end of total training epochs.

D Post-processing In Inference

In order to generate scores and singing voice in line with musical rules, we add some rule-based post-processing to the alignment predictions for more tolerance. For cases where total number of aligned notes is larger than the number of notes in the melody, we simply truncate the predicted number of aligned notes from the last token to the first or from the first token to the last. For cases of fewer number of predicted notes, we add the number of difference all to the last token.

E Data Annotation and Human Evaluation

Annotators are students who major in music, vocal singing or relevant specialty. They all speak bilingual languages with Chinese and English, so they are also qualified for translation quality evaluation. For data annotation and human evaluation, each person gets reasonably paid according to the
F  Song Translation Evaluations

Firstly, we clarify the signature\(^6\) for sacreBLEU in our evaluations. Notably, we take the lyrics that are both human-translated and can be sung as the single golden reference in our sacreBLEU score evaluations. Other human-translated lyrics are not guaranteed to be able to be sung. So the comparison of sacreBLEU scores including them may be unfair.

Besides, here we show the MOS-S and MOS-Q for Chinese-to-English song translation for reference. Lack of open data set for English singing voice synthesis caused the bad quality of synthesized English singing voice in inference. So we have to use the Chinese SVS system to synthesize the English songs. According to the feedback from annotators, this gap influence their feeling about translation results to some extent. As is shown in Table 4, results significantly drop compared to those of En→Zh. So we leave this to appendix part for reference.

<table>
<thead>
<tr>
<th></th>
<th>MOS-S</th>
<th>MOS-Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zh→En</td>
<td>4.05±0.08</td>
<td>4.06±0.07</td>
</tr>
<tr>
<td>GagaST</td>
<td>3.04±0.02</td>
<td>3.08±0.03</td>
</tr>
<tr>
<td>LTAG-cls</td>
<td>3.03±0.02</td>
<td>3.10±0.01</td>
</tr>
<tr>
<td>only bt</td>
<td>3.11±0.04</td>
<td>3.15±0.03</td>
</tr>
<tr>
<td>w/o bt</td>
<td>3.27±0.04</td>
<td>3.20±0.02</td>
</tr>
<tr>
<td>LTAG</td>
<td>3.39±0.03</td>
<td>3.61±0.05</td>
</tr>
<tr>
<td>only bt</td>
<td>3.39±0.05</td>
<td>3.60±0.02</td>
</tr>
<tr>
<td>w/o bt</td>
<td>3.29±0.05</td>
<td>3.32±0.02</td>
</tr>
</tbody>
</table>

Table 4: The Mean Opinion Score singability (MOS-S) and overall quality (MOS-Q) for Zh→En samples with 95% confidence intervals.

---

\(^6\)nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.0.0