INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Large Dataset Generation of Synchronized Music Audio and Lyrics at Scale using Teacher-Student Paradigm

Cristian Chivriga, Rinita Roy

Technical University of Munich, Germany
cristian.chivriga@tum.de, rinita.roy@tum.de

Abstract

Large models (e.g., GPT-3, CLIP, DALL-E) show remarkable few-shot and zero-shot capabilities when trained on hundreds of millions of samples. Despite this trend, no publicly available synchronized music audio and lyrics dataset of sufficient scale exists, nor does a reliable evaluation benchmark to assess a model's performance. To address this issue, we build and release *MusicLyric*, a large public dataset with over 320k audio sequences and lyrics pairs for a total duration of 1,200 hours based on a collection of over 32,000 songs. The generation process is based on the teacher-student paradigm where the student seeks to outclass the teacher with more data available using the newly generated pseudo-alignments. The method is efficient and straightforward with at least 3 iterations needed to create high-quality data that can be scaled to a hundred thousand samples. We make our dataset, toolkit, and pre-trained models open-source[1].

Index Terms: music information retrieval, dataset generation, automatic lyrics transcription, automatic lyrics alignment, speech recognition, non-autoregressive models

1. Introduction

Having large labeled training data is essential for training large models with high generalization and transfer capabilities. There has been considerable effort to generate such datasets [1, 2]. However, those are either limited in size or accessibility or are shallow by their properties, such as when the genre or the singer distribution is too skewed. For example, when the DALI dataset [1] was first released, it consisted of 5,358 audio tracks, only 2,748 of which are currently accessible via YouTube links as of October 2022. This amounts to merely 146 hours of training data. The toolkit was never made open-source, so it's not even possible to rebuild the dataset. Moreover, the genre distribution of the dataset is heavily skewed towards rock and pop music (1,804 and 1,509 tracks respectively), with other relevant genres being severely underrepresented (metal: 529, r&b: 159, hip-hop/rap: 64). There also exist datasets in the form of Karaoke interpretations (play-along) [2]. However, those datasets are still small ($\sim$150h [3]) and hardly accessible. To address this issue, we build and release a large public dataset with over 320k audio sequences and lyrics pairs for a total amount of 1,200 hours based on a collection of over 32,000 songs. We are going to describe the method to create such a dataset via the teacher-student paradigm similar to previous work [1]. Moreover, we justify the better characteristics of such a dataset compared to the existing ones. Furthermore, we demonstrate successful training with such datasets using state-of-the-art (SOTA) models for Automatic Speech Recognition (ASR), getting consistent results across all existing evaluation benchmarks.

[1] https://github.com/domainflag/music-lyric.

2. Methodology

We start by collecting over 32,000 audio and lyrics pairs (32,465 to be precise). We rely on highly popular charts such as Billboard charts[2] and extensive collections of audio metadata such as Million Song Dataset [4] for the extraction of song metadata (e.g., artist, title). Next, we perform filtering which is a two-step process consisting of finding the best candidate pair, lyrics content, and audio source.

2.1. Pre-processing

We use the YouTube API[3] to automatically query for a set of audio candidates that match some specific criteria, such as a minimum and maximum duration of 2 and 6 minutes respectively, a minimum view count of $1k$, and a rating score of 4.3 out of 5.0. We give higher priority to search results that include the keyword "lyrics" in the result title, as they are likely to contain audio materials without additional editing (interlude, promotional or explicit content). Due to limited storage capacity, we store the audio data in MP3 format with a low bit rate of 128 kbps, even though this may affect the ASR performance negatively [5].

For lyrics, we use multiple sources (e.g., *Genius*, *Musixmatch*, *LyricFind*) to increase the pool of high-quality matches for any audio source. We also filter out pairs that are not in English based on the transcript vocabulary. However, we allow a limited number of foreign words in the pre-processing stage. These words are well-represented and help increase the number of pairs available. The text pre-processing step involves removing noise (e.g., embedded tags, annotations, or URLs), converting numbers to words (e.g., "401" $\rightarrow$ "four-o-one" or "four hundred one"), transliterating foreign words (e.g., "Ros**é**" $\rightarrow$ "Ros**e**", "**Déjà** Vu" $\rightarrow$ "**Deja** Vu"), and normalizing non-lexical vocables (e.g., [**oo**h, **o**ooo**h**] $\rightarrow$ **oh**).

Initially, we selected the first pair of search results (i.e., lyrics content, and audio source) as the best match. In the final version, we used the best pre-trained teacher model from the former version to cherry-pick the best candidate pair among all query results with the lowest Character Error Rate (CER) score. That is opposed to using Word Error Rate (WER) which is less forgiving of errors while identifying words that are not in the standard vocabulary, such as slang, abbreviations, or neol-

[2] Billboard, https://www.billboard.com/, API https://github.com/guoguo12/billboard-charts.

[3] YouTube, https://www.youtube.com/, API https://github.com/ytdl-org/youtube-dl.

10.21437/Interspeech.2023-2174

ogisms. We distribute the dataset as URLs for lyrics and audio data. We have measures in place to mitigate accessibility issues, such as anti-blacklisting and automatic recovery mechanisms. For example, it can find alternative audio or lyrics sources if the original ones are down and realign them with the pre-trained teacher models.

2.2. Properties

MusicLyric dataset contains 32,465 samples with 7,063 unique artists; this number may vary as we do not impose any restrictions on the source material. For example, 198 tracks are from the comedy-drama television series "Glee", which features many song covers sung by different singers onscreen.

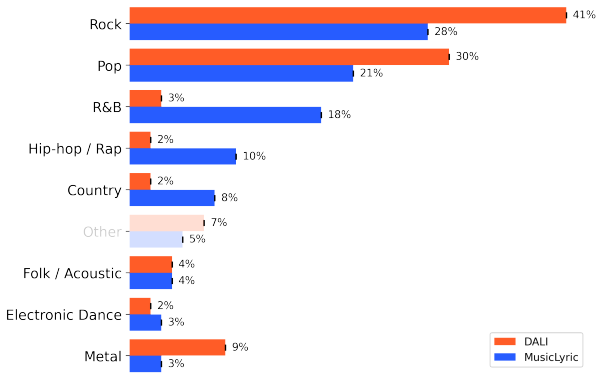

Figure 1: *Genre distribution: DALI [1] vs MusicLyric.*

Vocal music covers a wide range of styles that can be categorized by the music genre. For example, Hip-Hop music features rapping, which is a rapid, slangy, and rhythmic spoken delivery of rhymes and wordplay over background instrumentation with recurring beat patterns. In contrast, rock and metal are characterized by aggressive, raw, often raspy vocals that may involve screaming or growling paired with loud guitar riffs, aggressive drumming, and noise. According to the experiment conducted by Condit-Schultz and Huron [6], the intelligibility of lyrics varies with the style and to some extent with the genre. For instance, "Death Metal" pieces received zero intelligibility scores, i.e., no participant in the experiment could detect a single word. On the other hand, "Pop" excerpts achieved scores close to 100% due to the clean instrumentation and production anchored by clear vocal parts. Therefore, a rich and diverse collection of data should reflect a high diversity of music genres. Most of the songs' genre tags are extracted using the Last.FM API[4], which is a social music service that relies on community-driven tagging for accurate scoring of genre labels. Over 292 samples have insufficient tags because they are either newly released or covers or live versions of the official ones. The final genre distribution is shown in Figure 1. We extract the top 10 tags for each sample, collapse the sub-genres to their parent genres, and then choose the most common one. Compared to DALI [1], the distribution in *MusicLyric* is less skewed towards the "radio-friendly" music genres "Rock" and "Pop" that are highly accessible and popular among large audiences. On the other hand, some genres such as "Rap/Hip-Hop", "Country", and "R&B" are better represented; for example, there are 3,198 (10%) "Hip-Hop/Rap" samples in *MusicLyric* compared to only 64 (2%) in DALI. The minority consists of other genres, such as "Folk", "Jazz" and "Classical", that are less popular nowadays and harder to find with vocals. Overall, the actual distribution has less variance but also reflects better the listening patterns of large audiences by relying on popularity measures.

[4]Last.FM, `https://www.last.fm/`, API `https://github.com/feross/last-fm`.

2.3. Vocals Segmentation

Music samples are on average 3m 59s long, which makes learning the longer-term dependencies difficult and memory costly for end-to-end training. There have been considerable improvements by researchers in that respect [7, 8]. However, we perform chunking to ease future research development since the vocal parts make up 77.58% of the dataset's total duration (3m 05s per sample). We segment the vocals by using Spleeter [9] and perform clustering based on the mean amplitude over all frequency bands from the resulting Short-Time Fourier Transform (STFT) of the vocal audio signal. Subsequently, we select only the chunks containing vocals by greedily joining consecutive time steps whose energy exceeds a certain threshold (based on the previous split) with a maximum silence duration of 1s between them. Each chunk is restricted to be no more than 20 seconds long. We split the longer chunks by searching for the longest low-energy sequence as a simple heuristic method of detecting when a singer briefly catches their breath. While splitting, we also support padding by adding noise (instrumentation parts in the vicinity); otherwise, fixed boundaries for vocal parts will always be expected during training.

2.4. Alignment Framework

We explore a simple unsupervised generation method for precise alignments to create our dataset by designing a teacher-student alignment framework. We use multiple alignment methods in our model to assess the agreement over the generated alignments, including forced alignment through dynamic programming and beam alignment through anchoring. The former provides better alignment while the latter highlights any issues (e.g., multi-layered vocals) within the transcript, which we then fix. We include those pairs in the next training procedure if consensus emerges. Initially, the teacher is the pre-trained model AutoLyrixAlign [10] released open source, and the student is the model trained on pseudo-alignments generated by the teacher where we seek to outperform the teacher until little improvement is shown for cooperation. We evaluate the agreement between the two models by computing the WER between the region-based (chunk) transcription outputs given by them. The filtering threshold is set to 3%, but it can be adjusted with a trade-off between quality and quantity. Only the chunk transcript coming from the teacher's alignment is added to the pool of pairs D. There is a filtering stage before matching that involves discarding chunks coming from degenerate alignments (e.g., overlapping vocals) and a pre-processing stage that transforms the reference transcriptions L closer to the outputs T_t generated by a history of teacher models H_t. The set of operations consists of deletions, insertions, or substitutions required to fix L (e.g., chorus skipping, verse repetition) using a custom language model LM. In this iterative process, we re-train the model with newly acquired labeled data and reassess its cooperation (Refer to Figure 2). The alignment process takes less than 1 second per sample, and one whole iteration takes 14h to process 32,465 samples as we are constrained by the inference speed while using a single *P100* GPU.

Our end goal is to have a cooperation rate of 50%, which is accomplished by training more accurate systems with more

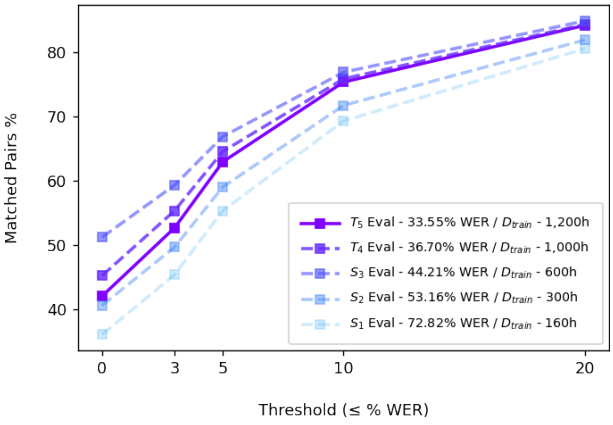

Figure 2: *Cooperation between AutoLyrixAlign[10] and multiple instances of our model Teachers (T) and Students (S).*

training data available. With only 3 iterations and a threshold of 0% WER, the matched pairs increased from 35.83% to 51.13% (Figure 2). First we seek cooperation between the open-source pre-trained model - AutoLyrixAlign [10] (teacher) and our student models (S_{1-3}). Eventually, our models rebel against the teacher as they get better, thus becoming teachers (T_{4-5}), and the pre-trained model becomes a student. This means that we can expand the dataset by one new sample for every two. We achieve a 76.73% match ratio only by using our latest teacher and student models. Key factors for improvement include the quality of reference lyrics transcripts, model performance in transcribing, alignment method, consensus method and aggregation method (e.g., majority vote). We also consider the previously trained teacher models using the majority vote aggregation method which increases the match ratio to 89.25%. A quick qualitative evaluation using our internal tool for a subset of randomly sampled 100 pairs shows zero label error with rare occasions of having at most one word mismatched. It is important to note that the number of perfectly matched whole pairs (not chunks) - 5,742 (17.68%) - is much lower due to frequent mismatches at the boundaries, where a whole sample has, on average, 12.53 chunks in aggregate. This level of granularity (i.e., chunking) allows us to harness a richer and wider pool of content with finer control over quality. Please see procedure 1 for more details about the method.

2.5. Evaluation Set

For evaluation purposes, we release a curated subset of 256 tracks that is further split into validation and test subsets. For simplicity's sake and to avoid blacklisting further samples for training, we re-use existing excerpts taken from available evaluation benchmarks [11, 12, 13, 14]. As described in subsection 2.2, we take a step further by considering candidates with multiple replicas (3) that are accessible via YouTube links with high longevity of at least two years. Additionally, we disregard samples using a much stricter and more complex filtering policy; for example, by avoiding non-fully matching lyrics per content, overlapping vocals, excerpts containing non-lexical vocables, the use of fade-out techniques while singing, and more. Models evaluated on songs by an artist encountered during training tend to perform exceptionally well, likely due to their familiarity with the artist's singing style and instrumentation. To prevent this bias, our train, validation, and test sets are disjoint with respect to the included artists. Additionally, both

Procedure 1: Dataset generation using Teacher-Student paradigm.

Data: N - Whole audio samples count - 32,000
L_N - Original pre-processed lyrics
C_N - Where $c_i \in C$ is a set of vocal chunks

Initialize teacher model $T \leftarrow AutoLyrixAlign$;
History of teacher models $H_t \leftarrow \emptyset$;
Track the 3rd best model $M_p \leftarrow null$;
Generate initial dataset D given T and C;
while *True* **do**
  Train our own new student model S on D;
  Compare S, T, M_p on the test set;
  if M_p *is not null* ***and*** M_p *is better than* S **then**
    Continue; *(e.g., underfitting → architecture refinement or increase model complexity)*
  else if S *is better than* T **then**
    $H_t \leftarrow Union(H_t, T)$;
    $T, S \leftarrow S, T$;
  $L_s \leftarrow Transform(L|H_t, LM)$;
  $D \leftarrow \emptyset$;
  Generate labels T_t and S_t;
  Create alignments T_a and S_a given $[T, S]_t$ and L_s;
  foreach $c_i \in C$ **do**
    $score \leftarrow WER(L_s[c_i|T_a], L_s[c_i|S_a])$;
    if $score \leq 3\%$ **then**
      $D \leftarrow Union(D, L_s[c_i|T_a])$;
  $M_p \leftarrow S$;

the validation and the test sets include no more than one song per artist. Lastly, we consider a balanced distribution over the singers' gender, release time (for a total of five decades starting from the 70s), and music genre (based on eight parent genres). We acknowledge that accessibility issues might still persist. If a reference gets invalidated, we can't fully guarantee perfect recovery (i.e., verse missing). However, over one year of ongoing research, we had zero loss of information on the validation and test set and minimal changes to the train set with no impact on its quantity or quality.

3. Model

We use the SOTA model, Conformer [15] (L), for training on the newly curated dataset. Additionally, we use a custom transformer-based architecture [16] adapted for the task of training on noisy data to train on our music data. The inputs to both networks are a sequence of 80-channel filterbanks features computed by a fixed window of 25ms with a stride of 10ms. Initially, the input features are both normalized to zero mean and unit variance. We use SpecAugment [17] with a mask parameter F = 27, a slightly lower time masking frequency m_T = 4 [15], and a time mask size depending on the input size r = 0.05. For a fair comparison, both models use the same convolutional feature encoder [15]. Additionally, the Conformer has an extra layer as compensation for replacing the LSTM decoder with a linear one. Both models share the same number of parameters with the same number of layers (18 layers).

4. Training

Both models used are trained on the *MusicLyric* dataset (1,200) in a non-autoregressive fashion using a Connectionist Temporal Classification (CTC) [18] decoder for cheaper experimentation. We favor higher granularity at decoding, instead of using a sub-word vocabulary; otherwise, the alignment quality degrades drastically with a higher mismatch error rate at the

boundaries. The output prediction space contains the English alphabet, space, apostrophe, and blank (special character for CTC). During training, we enable batch bucketing based on the length of the labels with much better generalization, observed empirically. We train both models for 100k optimization steps with Adam optimizer [19] that has an initial learning rate 1×10^{-4} with default $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We linearly decay the learning rate until it reaches the lowest of 4×10^{-5}. The batch size is set to 24 for faster convergence, and we use native mixed precision training on a single GPU to reduce memory footprint. Additionally, we continue training for another $10k$ steps using a larger batch size (2048) via gradient accumulation. This reduces the variance coming from mini-batch training and improves the performance by reaching a quasi-optimal solution with finer predictions. While training exclusively on the *MusicLyric* dataset, we find it extremely helpful to temporarily disable augmentation and warm up the learning rate for the first $24k$ steps. Since the dataset has a high variance, we hypothesize that the momentum-based optimizers stray badly into inaccurate search directions if the initial learning rate is too high or if too much augmentation is applied. A better alternative is to perform knowledge transfer from similar tasks (e.g., LibriSpeech [20]) by pre-training for a few epochs. This avoids poor initialization of the attention mechanism. We only apply pre-training while training new models for dataset generation. We also use a fixed dropout probability ($p = 0.1$) for all our models. The overall training time is approximately 15.3 days on the whole dataset on a single P100 GPU. While evaluating, we consider a single type of language model for acoustic model decoding: a 4-gram model using KenLM toolkit [21] which is trained on the corpus based on the train-set data. We use a small budget of 10 trials, each with a beam-width of 200, to estimate the best hyperparameters ($\alpha = 0.69$, $\beta = 1.68$) while decoding using CMA-ES sampler [22, 23].

5. Experiments

First, we ask ourselves whether background music paired with clean vocals is worth consideration while training our models on music data. Spleeter [9] adds distortion to the extracted singing vocals having some sort of muffled sound to it. That can adversely affect the performance of our model and possibly far outweigh the negative effect of having noisy instrumentation while training on polyphonic music. As shown in Table 1, we see no improvement when the models are trained on original polyphonic audio as opposed to what was shown by Gupta et al. [10]. The model trained on vocals performs generally better at lyrics transcription, although it comes with the cost of having alignment of worse quality. For further evaluation, however, we still only consider the model trained on vocals, as it achieves the lowest transcription error and outperforms the Conformer [15] architecture.

Table 1: *Evaluation results — ours vs Conformer [15] — for MusicLyric dataset (polyphonic audio and vocals only).*

Model	Dataset Type	WER (%)
Ours	Polyphonic Audio	32.13
	Vocals Only	**31.74**
Conformer [15]	Polyphonic Audio	35.05
	Vocals Only	34.62

Table 2: *Evaluation (WER%) on four music-related evaluation sets for lyrics transcription.*

Model	Jamendo	Mauch	Hansen	Dali-test
DDA [24]	72.15	75.39	74.81	-
CG [10]	59.60	44.00	-	-
GGL [25]	56.76	43.76	45.88	-
MSTRE-Net [11]	34.94	37.33	36.78	42.11
E2E Trans. [3]	44.12	33.69	36.85	40.20
PoLyScriber [26]	41.00	32.83	33.57	36.52
TransferLearn [27]	33.13	28.48	**18.71**	30.85
Ours	**28.74**	**27.86**	28.98	**28.60**

We also compare our model with those of existing SOTA approaches for lyrics transcription on the available benchmarks and use the findings as a proxy for whether our dataset is of sufficient quality. To assess the quality of our dataset, we evaluate the performance of our model for lyrics transcription on available benchmarks: Mauch[5] [14] (20 songs), Jamendo [12] (20 songs), Hansen [13] (9 songs) and DALI-test [11] (240 songs).

Having the same model trained separately on randomly sampled, same-sized subsets of the *MusicLyric* and other datasets [1, 2] would best justify our claims. Unfortunately, those sets are restricted, and our many access requests have been unsuccessful. Despite this, we consistently perform well on all existing evaluation benchmarks for lyrics transcription and achieve SOTA results as shown in Table 2, which serves as the closest proxy for the previously mentioned efforts. This can be attributed to the nature of our dataset incorporating diverse music styles. However, further analysis is required to validate that statement. It should be noted that our model does not adopt complex music-specific training strategies that might be responsible for even higher transcription performance. For future work, we can consider various strategies such as joint E2E vocal extraction and lyrics transcription [26], genre-informed acoustic model training [25], and music tagging [11]. Additionally, we would expect even higher performance gains by switching to a Seq2Seq architecture, that works similar to how humans depend on contextual information. This is especially true when they have to distinguish non-intelligible, highly ambiguous spoken words found in songs.

6. Conclusion

In this paper, we presented a framework that compares the transcriptions generated by the teacher and student models with the reference lyrics content and produces accurate time-aligned labels for the melody. We also released a large-scale, high-availability dataset for music and lyrics which is fully open to the broad community. We showed that the dataset is of sufficient quality by achieving SOTA results and performing consistently well across all the existing evaluation benchmarks using SOTA models for ASR. The dataset has less skewness and comes with easy-to-use tools for development and debugging. Our work can be used for various tasks including lyrics transcription, music matching [28], audio-to-lyrics alignment, dataset expansion, and supervision or evaluation via our bigger and richer set. This work lays the foundation for large-scale dataset generation and future research on E2E language-acoustic models, not restricted by the limited data size or accessibility issues.

[5]For Mauch [14], we are missing two samples, specifically "Once In A Lifetime" and "Someday" by Shinya Iguchi (RWC).

7. References

[1] G. Meseguer-Brocal, A. Cohen-Hadria, and G. Peeters, "Dali: a large dataset of synchronized audio, lyrics and notes, automatically created using teacher-student machine learning paradigm." in *19th International Society for Music Information Retrieval Conference*, ISMIR, Ed., September 2018.

[2] G. R. Dabike and J. Barker, "Automatic lyric transcription from karaoke vocal tracks: Resources and a baseline system," in *INTERSPEECH*, 2019.

[3] X. Gao, C. Gupta, and H. Li, "Automatic lyrics transcription of polyphonic music with lyrics-chord multi-task learning," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 2280–2294, 2022.

[4] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere, "The million song dataset," in *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*, 2011.

[5] M. Borský, P. Mizera, P. Pollák, and J. Nouza, "Dithering techniques in automatic recognition of speech corrupted by mp3 compression: Analysis, solutions and experiments," *Speech Commun.*, vol. 86, pp. 75–84, 2017.

[6] N. Condit-Schultz and D. Huron, "Catching the lyrics: Intelligibility in twelve song genres," *Music Perception: An Interdisciplinary Journal*, vol. 32, pp. 470–483, 2015.

[7] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. Le, and R. Salakhutdinov, "Transformer-XL: Attentive language models beyond a fixed-length context," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 2978–2988. [Online]. Available: https://aclanthology.org/P19-1285

[8] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The long-document transformer," *CoRR*, vol. abs/2004.05150, 2020. [Online]. Available: https://arxiv.org/abs/2004.05150

[9] R. Hennequin, A. Khlif, F. Voituret, and M. Moussallam, "Spleeter: a fast and efficient music source separation tool with pre-trained models," *Journal of Open Source Software*, vol. 5, no. 50, p. 2154, 2020, deezer Research. [Online]. Available: https://doi.org/10.21105/joss.02154

[10] C. Gupta, E. Yılmaz, and H. Li, "Automatic lyrics alignment and transcription in polyphonic music: Does background music help?" in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 496–500.

[11] E. Demirel, S. Ahlbäck, and S. Dixon, "Mstre-net: Multistreaming acoustic modeling for automatic lyrics transcription," in *International Society for Music Information Retrieval Conference*, 2021.

[12] D. Stoller, S. Durand, and S. Ewert, "End-to-end lyrics alignment for polyphonic music using an audio-to-character recognition model," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 181–185.

[13] J. K. Hansen, "Recognition of phonemes in a-cappella recordings using temporal patterns and mel frequency cepstral coefficients," in *Ninth Sound and Music Computing Conference (SMC)*, 2012.

[14] M. Mauch, H. Fujihara, and M. Goto, "Integrating additional chord information into hmm-based lyrics-to-audio alignment," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, pp. 200–210, 2012.

[15] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented Transformer for Speech Recognition," in *Proc. Interspeech 2020*, 2020, pp. 5036–5040.

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[17] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *Interspeech 2019*, 9 2019. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-2680

[18] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," vol. 2006, 01 2006, pp. 369–376.

[19] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015.

[20] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An asr corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.

[21] K. Heafield, I. Pouzyrevsky, J. Clark, and P. Koehn, "Scalable modified kneser-ney language model estimation," in *ACL*, 2013.

[22] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 2623–2631. [Online]. Available: https://doi.org/10.1145/3292500.3330701

[23] N. Hansen, "The CMA evolution strategy: A tutorial," *CoRR*, vol. abs/1604.00772, 2016. [Online]. Available: http://arxiv.org/abs/1604.00772

[24] E. Demirel, S. Ahlbäck, and S. Dixon, "Low resource audio-to-lyrics alignment from polyphonic music recordings," *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 586–590, 2021.

[25] X. Gao, Y. Yang, C.-C. Lee, and H. Li, "Lyrics transcription and lyrics-to-audio alignment with music-informed acoustic models," in *2020 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2020, p. 1019–1026.

[26] X. Gao, C. Gupta, and H. Li, "Polyscriber: Integrated training of extractor and lyrics transcriber for polyphonic music," 2022. [Online]. Available: https://arxiv.org/abs/2207.07336

[27] L. Ou, X. Gu, and Y. Wang, "Transfer learning of wav2vec 2.0 for automatic lyric transcription," *arXiv preprint arXiv:2207.09747*, 2022.

[28] R. Roy, R. Mayer, and H.-A. Jacobsen, "Mdms: Music data matching system for query variant retrieval," in *Proceedings of the 29th ACM International Conference on Multimedia*, ser. MM '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 2762–2764. [Online]. Available: https://doi.org/10.1145/3474085.3478551