MINING LABELED DATA FROM WEB-SCALE COLLECTIONS FOR VOCAL ACTIVITY DETECTION IN MUSIC

Eric J. Humphrey[1] **Nicola Montecchio**[1] **Rachel Bittner**[1,2]
Andreas Jansson[1,3] **Tristan Jehan**[1]
[1] Spotify, New York, USA
[2] Music, Audio & Research Lab (MARL), New York University, USA
[3] City University, London, UK
{ejhumphrey, venice, rachelbittner, andreasj, tjehan}@spotify.com

ABSTRACT

This work demonstrates an approach to generating strongly labeled data for vocal activity detection by pairing instrumental versions of songs with their original mixes. Though such pairs are rare, we find ample instances in a massive music collection for training deep convolutional networks at this task, achieving state of the art performance with a fraction of the human effort required previously. Our error analysis reveals two notable insights: imperfect systems may exhibit better temporal precision than human annotators, and should be used to accelerate annotation; and, machine learning from mined data can reveal subtle biases in the data source, leading to a better understanding of the problem itself. We also discuss future directions for the design and evolution of benchmarking datasets to rigorously evaluate AI systems.

1. INTRODUCTION

Over the last few years, the ubiquity of cheap computational power and high quality open-source machine learning software toolkits has grown considerably. This trend underscores the fact that attaining state-of-the-art solutions via machine learning increasingly depends more on the availability of large quantities of data than the sophistication of the approach itself. Thus, when tackling less traditional or altogether novel problems, machine learning practitioners often choose between two paths to acquiring data: manually create (or curate) a dataset, or attempt to leverage existing resources.

Both approaches present unique challenges. Curation is necessary when precise information is required or insufficient data are available, but can incur large costs in both time and money. Alternatively, "mining" data – recovering useful information that occurs serendipitously in different contexts – can result in large datasets with far less effort, *e.g.,* recovering labels from the text around an image. While this information is typically generated as a by-product of some other pre-existing human activity and prone to both noise and bias, recent machine learning research has managed to use this approach to great effect [5].

With the continued growth of digital music services, vocal activity detection (VAD) is a task of increasing importance. Robust VAD is a key foundational technology that could power or simplify a number of end-user applications, such as vocalist similarity, music recommendation, artist identification, source separation, or lyrics transcription. Despite previous research, the state of the art continues to advance with diminishing returns, rendering VAD an unsolved problem with considerable potential.

Given the dominance of data-driven methods in machine learning, it stands to reason that data scarcity may be contributing to the apparent ceiling in the performance of VAD algorithms. Modest progress has been made toward increasing the size of labeled datasets, limiting the efficacy of modern approaches, *e.g.,* deep learning. Efforts to leverage strongly labeled datasets have converged to hundreds of observations [1,13,15,16], with which complex methods have been explored [9, 10, 18]. Recent research succeeded in curating a private dataset of $10k$, 30 second weakly labeled clips, *e.g.,* "completely instrumental" or "contains singing voice", using this dataset to train convolutional neural networks [17].

In short, VAD research remains largely dependent on sustained human involvement in sourcing labeled data, but this approach struggles to scale. Here, we propose leveraging a huge, untapped resource in modern music to circumvent this challenge: the "instrumental version", *i.e.,* a song in which the vocals have been omitted. The goal of this work is thus the exploration of this opportunity, achieved in four steps: mine original-instrumental pairs from a massive catalogue of music content; estimate time-varying vocal density given corresponding tracks; exploit this signal to train deep neural networks to detect singing voice; and understand the effects of this data source on the resulting models.

© Eric J. Humphrey, Nicola Montecchio, Rachel Bittner, Andreas Jansson, Tristan Jehan. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Eric J. Humphrey, Nicola Montecchio, Rachel Bittner, Andreas Jansson, Tristan Jehan. "Mining Labeled Data from Web-Scale Collections for Vocal Activity Detection in Music", 18th International Society for Music Information Retrieval Conference, Suzhou, China, 2017.

2. DATA GENERATION

In Western popular music, a song's arrangement often revolves around a lead vocalist, accompanied by instruments such as guitar, drums, bass, *etc.* It is not uncommon for an artist to *also* release an "instrumental" version of the same song (to be used for *e.g.,* remixes or karaoke), in which the primary difference between it and the corresponding "original" recording is the absence of vocals. [1] In principle, the difference between these two sound recordings should be highly correlated with vocal activity, which would provide a fine-grained signal for training machine learning models. However, to exploit this property at scale, it is necessary to identify and align pairs of original recordings and matching instrumental versions *automatically*.

We outline a three-step approach toward mining strongly labeled instances of singing voice from a music catalogue: identify original-instrumental pairs from track metadata; estimate a vocal density signal for the original track, given its instrumental; draw positive observations from an original track as a function of estimated vocal density.

2.1 Selection of Matching Recordings

We search the full Spotify catalogue, a set of tens of millions of commercially recorded tracks, for paired versions using a heuristic based on track metadata. A pair of tracks (A, B) are marked as $(\text{original}, \text{instrumental})$ if:

- A and B are recorded by the same artist.
- "instrumental" does not appear in the title of A.
- "instrumental" does appear in the title of B.
- The titles of A and B are *fuzzy matches*.
- The track durations differ by less than 10 seconds.

Fuzzy matching is performed on track titles by first latinizing non-ASCII characters, removing parenthesized text, and finally converting to lower-case; this yields about 164k instrumental tracks. Note that this is a many-to-many mapping, as an original version can point to several different instrumentals, and vice versa.

A tiny subset of this content is manually reviewed to check for quality, and we find roughly 1 in 10 tracks to be a mismatched pair: the majority of errors are due to instrumental tracks that appear on multiple albums, such as compilations or movie soundtracks, but are only tagged as such in some contexts. An open-source audio fingerprinting algorithm is used to remove suspect pairs from the candidate set [6]. Sequences of codes for tracks are extracted, and track pairs are discarded as a function of Jaccard similarity if code sequences do not overlap sufficiently (an erroneous fuzzy metadata match) or overlap too much (the tracks in the pair being both instrumental or vocal). Finally, redundant associations from this mapping are removed, so that each original track is linked to only one instrumental track. Overall this process yields roughly $24k$ tracks, or $12k$ original-instrumental pairs, totaling some 1500 hours of audio.

2.2 Estimating Vocal Density

Let T^O and T^I denote two recordings, corresponding to an "original" and "instrumental" version, respectively. A Time-Frequency Representation (TFR) is computed for both tracks, respectively X^O and X^I. Subsequently, the TFRs are aligned to estimate time-varying vocal density.

In this work a *Constant-Q Transform* (CQT) [3] is chosen for its complementary relationship between convolutional neural networks and music audio; the CQT uses a logarithmic frequency scale that linearizes pitch, allowing networks to learn pitch-invariant features as a result [8]. The frequency range of the transform is constrained to the human vocal range, *i.e.,* E2 - E7 (5 octaves, spanning 82.4-2637 Hz), and a moderately high resolution is employed, with 36 bins per octave and 32 frames per second. Logarithmic compression is applied pointwise to the TFR.

The pair of TFRs (X^O, X^I) undergoes a feature dimensionality reduction via Principal Component Analysis [2], producing (Z^O, Z^I); based on empirical findings, $k = 20$ components were found to yield good results. This step not only provides an increase in computational efficiency in subsequent processing steps, but also affords a useful degree of invariance because of the lower dimensionality.

The transformed sequences are then aligned using Dynamic Time Warping (DTW), yielding two sequences, n^O, n^I, of indices over the original and instrumental song, respectively [14]. This allows us to recover points in time from both a full and instrumental mix where the background musical content is roughly identical.

Using these indices, the CQT spectra (X^O, X^I) are resampled to equivalent shapes, and the half-wave rectified difference between log-magnitude spectra yields the following residual:

$$\delta_{j,k} = \max\left(0, \log|X^O_{n^O_{j,k}} + 1| - \log|X^I_{n^I_{j,k}} + 1|\right) \quad (1)$$

In the ideal case, where any difference is due entirely to vocals, this residual represents the vocal CQT spectra, and will behave like a smooth contour through successive time-frequency bins. Practically, however, there will likely be other sources of residual energy, due to suboptimal alignment or true signal differences. To best characterize contour-like residuals, we normalize the spectral energy in each time frame and apply the Viterbi algorithm to decode the most likely path through the residual spectra; this step is inspired by previous work on tracking fundamental frequency in a time-frequency activation map [12]. Empirically we find this process far more robust to residual noise than simpler aggregation schemes, such as summing energy over frequency.

[1] Though other differences in signal characteristics may occur due to production effects, *e.g.,* mastering, compression, equalization, these are not considered here.

[2] We are not interested in learning a general transform; the Principal Components of each pair of tracks are computed independently of the overall dataset.

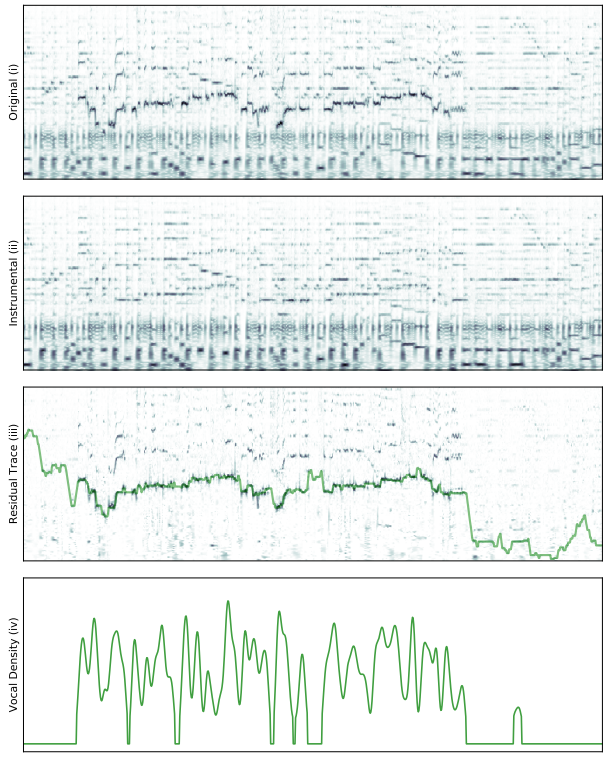

Figure 1. The intermediate stages in estimating vocal density from an original-instrumental pair of recordings, showing (i) the original and (ii) instrumental CQT spectra, (iii) the residual with a trace of its fundamental, and (iv) the estimated vocal density over time.

The amplitude of this time-frequency path, ρ, provides an estimate of vocal *density*, ϕ, the likelihood that vocals are present in the *original* recording, T^O, as a function of time. Finally, we forwards-backwards filter ϕ with a Hanning window ($L = 15$), to both smooth and dilate the density signal to encompass vocal onsets and offsets. The stages of this process are pictured in Figure 1.

2.3 Sampling of Positive and Negative Observations

Having estimated *where* vocals likely occur in a piece of audio, we turn our attention to *how* this information is utilized for supervised training. We highlight that track-level metadata presents a multiple-instance learning problem, where each recording can be understood as a *bag* of samples with a single binary label: "vocal" if it contains at least one positive sample, or "non-vocal" if all samples are negative. In this setting, positively labeled bags are inherently noisy, with some unknown percentage of negative samples effectively mislabeled as a result. To address this issue, the vocal density estimate is used to reweight the contributions of samples drawn from positive bags.

An estimator is trained by drawing positive ($Y = 1$) and negative ($Y = 0$) samples from original and instrumental tracks, with equal frequency. Negative samples are drawn uniformly from instrumental tracks, while positive samples are drawn as a function of the vocal density ϕ. To smoothly interpolate between a uniform distribution and the vocal density estimate over positive samples, two parameters are introduced, a threshold, τ, and a compression factor, ϵ:

$$Pr(X_n^O|Y=1) \propto \begin{cases} \phi_n^\epsilon & \phi_n \geq \tau \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

Here, we are interested in exponentials in the range of $0 < \epsilon < 1$, which flatten the density function. Note that $\epsilon = 0, \tau = 0$ corresponds to uniform sampling over time, and is equivalent to the weakly labeled setting, *i.e.,* a label applies equally to all samples.

As a final consideration, we highlight that the original and instrumental recordings are aligned in the course of computing the vocal density estimate. Therefore, it is possible to draw correlated positive-negative pairs from both the original and instrumental tracks corresponding to the same point in time, a sampling condition we refer to as *entanglement*, $\zeta \in \{True, False\}$. One would expect that these paired samples live near the decision boundary, being near-neighbors in the input space but belonging to different classes, and we are interested in exploring how training with entangled pairs may affect model behavior.

3. SYSTEM DESIGN

3.1 Previous Approaches

The majority of VAD research follows a similar architecture: short-time observations are fed to a classifier, each observation is assigned to either a vocal or a non-vocal class, and optionally post-processing is applied to eliminate spurious predictions. Early work uses "the acoustic classifier of a speech recognizer as a detector for speech-like sounds" to feed an Artificial Neural Network trained on a speech dataset (NIST Broadcast News) [1], while [16] attempts to explicitly exploit *vibrato* and *tremolo*, two characteristics that are specific of vocal signals. Alternatively, Support Vector Machines (SVMs) are used for frame classification and Hidden Markov Models act as smoothing step [15]; a similar solution is proposed by [13], which exploits a wider set of features, including ones derived from a predominant melody extraction step.

More recently, increasingly complex classifiers are preferred to feature engineering, given the widespread success of deep learning methods and modest increases in available training data. Much prior research explores the application of deep learning to music tagging, which typically encompasses one or more classes for singing voice in the taxonomy considered [7]. Elsewhere, deep networks have been used for pinpointing singing voice in source separation systems [19]. Regarding the particular task at hand, [9] proposes a sophisticated architecture based on Recurrent Neural Networks that does not have a separate smoothing step, while [17] uses a conventional convolutional network topology, further advancing the state of the art.

3.2 Proposed System

The log-magnitude CQT representation described in 2.2 is processed in 1 second windows, with a dimensional-

ity of (32×180) bins in time and frequency, respectively. We adopt a five-layer neural network, with three (3D) convolutional layers, each followed by max-pooling, and two fully-connected layers, with the following parameter shapes: $w_0 = (1, 64, 5, 13), p_0 = (2, 3)$, $w_1 = (64, 32, 3, 9), p_1 = (2, 2), w_2 = (32, 24, 5, 1), p_2 = (2, 1)$, $w_3 = (1540, 768), w_4 = (768, 2)$. All layer activations are hard rectified linear units (ReLUs), with the exception of the last (classifer) layer, which uses a softmax.

The network is trained using a negative log-likelihood loss function and parameters are optimized with minibatch stochastic gradient descent. We implement our model in *Theano* [3], leveraging the *Pescador* [4] package for drawing samples from our dataset, and accelerate training with a NVIDIA Titan X GPU. Networks are trained for 500k iterations ($\approx$ 20 hours) with a learning rate of 0.05 and a batch size of 50. Dropout is used in all but the last layer, with a parameter of 0.125. In addition to the weakly labeled case, $\{\epsilon = 0.0, \tau = 0.0, \zeta = F\}$, we explore model behavior over two sampling parameter settings, with and without entanglement: $\{\epsilon = 0.3, \tau = 0.05\}$ and $\{\epsilon = 1.0, \tau = 0.2\}$. These values are informed by first computing a histogram of vocal activation signals over the collection, revealing that a large number of values occur near zero ($\leq$ 0.05), while the upper bound rolls off smoothly at ≈ 2.5.

4. EXPERIMENTAL RESULTS

We evaluate our models on two standard datasets in VAD research: the *Jamendo* collection, containing 93 manually annotated songs [15]; and the *RWC-Pop* collection, containing 100 manually annotated songs [13]. To compare with previously reported results, we consider the area under the curve (AUC) score and max-accuracy [17]. The AUC score provides insight into the rank ordering of class likelihoods, and max-accuracy indicates the performance ceiling (or error floor) given an optimal threshold.

4.1 Quantitative Evaluation

Table 1 shows the summary statistics over the two datasets considered as a function of sampling parameters, alongside previously reported results for comparison [17]. The first three systems (α, β, γ) are successive boosted versions of each other, *i.e.,* α is trained with weak labels, and its predictions on the training set are used to train β, and so on; the *fine* model is trained directly with strongly labeled data, and we refer to each by suffix, *e.g.,* α. Additionally, the authors train these models with a witheld dataset unavailable to our work here.

These results provide a few notable insights. First, we confirm that our automated approach of mining training data is sufficient to train models that can match state of the art performance. Configuration `I`, corresponding to the weak labeling condition, performs roughly on par with a comparably trained system, α, validating previous results; configuration `V` achieves the best scores of our models, and

[3] `https://github.com/Theano/Theano`
[4] `https://github.com/pescadores/pescador`

				RWC		JAMENDO	
				AUC	ACC$^+$	AUC	ACC$^+$
SCHLÜTER-α				0.879	0.856	0.913	0.865
SCHLÜTER-β				0.890	0.861	0.923	0.875
SCHLÜTER-γ				0.939	0.887	**0.960**	**0.901**
SCHLÜTER-FINE				**0.947**	0.882	0.951	0.880
	τ	ϵ	γ				
I	0.0	0.0	F	0.891	0.856	0.911	0.856
II	0.05	0.3	F	0.918	0.879	0.925	0.869
III	0.05	0.3	T	0.918	0.879	0.934	0.874
IV	0.2	1.0	F	0.937	0.887	0.935	0.872
V	0.2	1.0	T	0.939	**0.890**	0.939	0.878

Table 1. AUC-scores and maximum accuracies across models on the *RWC* and *Jamendo* datasets.

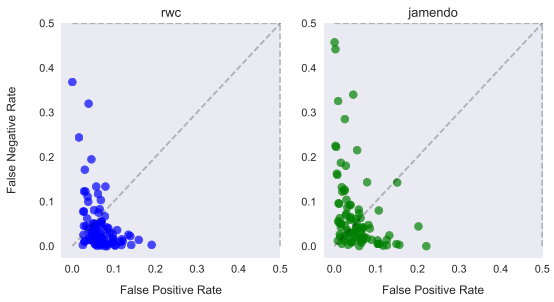

Figure 2. Trackwise error rates, plotting false positives versus false negatives for IV; one outlier ($fn \approx 0.66$) in the *Jamendo* set is not shown to maintain aspect ratio.

is consistent with gains in prior work. That said, the difference between models is in the range of 0.02-0.05 across metrics, which is of limited reliability with datasets of this size. In terms of sampling parameters, we observe a direct correlation between signal-to-noise ratio in our training data, *i.e.,* the more non-vocal observations are discarded, the better the models behave on these measures. Training with entangled pairs ($\zeta = T$) also seems to have a small positive effect. Finally, we note a possible corpus effect between these systems and previously reported results, where models (V and γ) perform better on different data. Though minor, this potential corpus effect serves as a dimension to explore in subsequent analysis.

4.2 Error Analysis

As the systems reported here are high performing, a potentially more informative path to understanding model behavior is through analyzing the errors made. Here, false positives occur when a different sound source is mistaken for voice; false negatives occur when the energy of a vocal source has fallen below the model's sensitivity. Observations drawn from the same music recording will be highly correlated, due to the repetitive nature of music, and so we explore the track-wise frequency of errors to identify behaviors that may reveal broader trends.

A slight corpus effect is seen in Figure 2 between the *RWC* and *Jamendo* collections. In the former, the majority

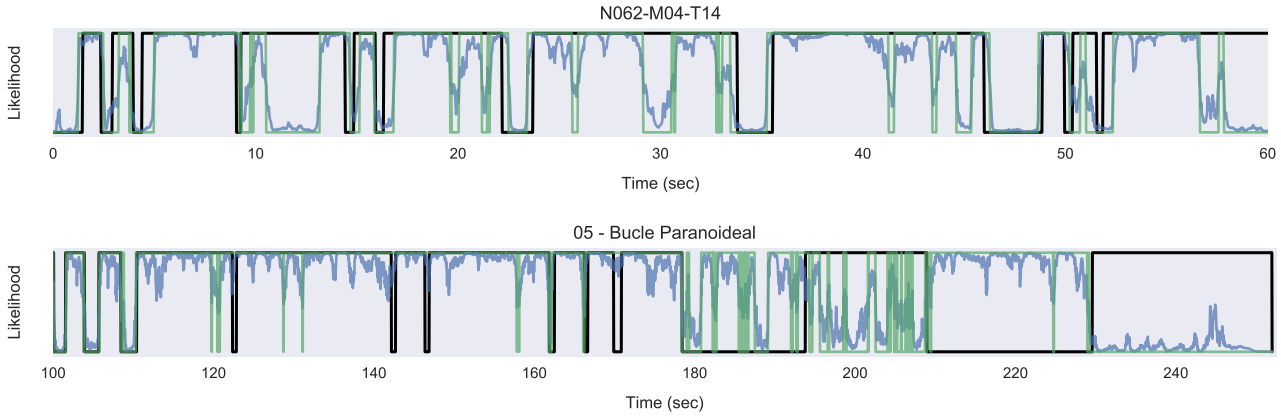

Figure 3. Examples from the evaluation dataset, showing the ground truth (black), estimated likelihoods (blue) and thresholded prediction (green) over time: (top) a track from the *RWC* corpus demonstrates how a model can operate with greater temporal precision than a human annotator, a common source of false negatives; (bottom) a track from the *Jamendo* collection illustrates different challenges, including imposter sources (a guitar solo), sensitivity to vocals, and annotator error.

of error is due to false positives, but at a much lower rate of occurrence ($fp < 0.1$) than false negatives. Additionally, when errors do occur in a track, they tend to be primarily of one type, and seldom both. This is less the case for the *Jamendo* set, comprised of both "worse" tracks and a (slightly) larger co-occurrence of error types in a given track.

Using this visualization of trackwise errors, an investigation into the various outliers yields a few observations. There are two primary sources of false negatives: one, shown in Figure 3 (top), trained models exhibit a level of temporal precision beyond the annotators' in either dataset, pinpointing breaths and pauses in otherwise continuous vocalizations; and two, nuances of the data used for training seem to induce a production bias, whereby the model under-predicts singing voice in lower quality mixes. In hindsight, it is unsurprising that models trained on professionally produced music might develop a sensitivity to mixing quality, and we note this as a topic for future exploration. A similar bias also appears to account for the majority of all false positives, often corresponding with monophonic instrumental melodies, *e.g.,* guitar riffs or solos, but less so for *polyphonic* melodies, *i.e.,* two or more notes played simultaneously by the same source, consistent with previous findings [11].

Figure 3 (bottom) illustrates an interesting example of this behavior. In the first 80 seconds shown here, the model agrees with the human annotation. The model fails at the 180 second mark, misclassifying a guitar line, and continues through 194-210 seconds, where the model struggles to detect rap vocals at a softer volume. However, from that point onwards, the human annotation itself is wrong, while the model is correct; vocals are indeed present between 210-230 seconds and 230-252 contains no voice, which accounts for 16% of the track. Coincidentally, this further underscores the need for large, diverse evaluation datasets to produce reliable metrics.

4.3 Multitrack Analysis

The above results confirm the intuition that it can be challenging to manually annotate singing voice activity with machine precision. Ideally, though, human annotation approximates a smoothed, thresholded version of the vocal signal energy in isolation, and as such, we are interested understanding the degree to which model estimations correspond with the pure vocal signal. Another way of measuring our models' capacity to estimate singing voice from a "down-mixed" recording is through the use of multitrack audio, which provides direct access to the signal of interest, *i.e.,* vocals, in isolation.

We now turn our attention to MedleyDB, a dataset of 122 songs containing recordings of individual stems and corresponding mixes [2]. For each of the 47 songs that have vocals in isolation, we create a single vocal track for analysis, and compute the log-magnitude CQT for the full mix (the "original" version) X^M, and the isolated vocals, X^V. Whereas previously Viterbi was used to track vocal density, here the reference vocal density contains no noise and can be computed by summing the spectral energy over frequency, *i.e.,* $\phi_n^V = \sum_k X_{n,k}^V$. The trained models are applied to the full mix, X^M, for inference, producing a time-varying likelihood, L^M.

The reference vocal density is not a class label but a continuous value, and the comparison metrics must be adjusted accordingly. Maximum accuracy is generalized to the case where independent thresholds are considered for ϕ^V, L^M over the dataset, providing insight into the best-case agreement between the two signals. We also consider the Spearman rank-order correlation between the two sets, a measure of the relative rank order between distributions [20].

An exploration of model performance on this dataset validates earlier observations, summarized in Table 2. On manual inspection of the temporal precision of the model on the Medley dataset, we confirm that deviations between estimated likelihoods and the reference vocal density are

	$\{\tau, \epsilon, \zeta\}$	SPEARMAN-R	ACC$^+$
I	0.0, 0.0, F	0.681	0.812
II	0.05, 0.3, F	0.779	0.849
III	0.05, 0.3, T	0.768	0.854
IV	0.2, 1.0, F	0.784	0.852
V	0.2, 1.0, T	**0.796**	**0.862**

Table 2. Spearman rank-order correlation and maximum accuracy scores across models on the MedleyDB vocal subset.

representative of true model errors, setting a baseline for future work. As seen previously, false negatives again correspond to vocal loudness relative to the mix, and false positives are caused by loud melodic contours. Note also that the Spearman rank-order correlation is consistent with previously observed trends across models, while providing more nuance; the greatest difference between models is > 0.11, versus ≈ 0.05 for maximum accuracy. Finally, we note that the flexibility of multitrack datasets presents a great opportunity for rigorously testing future work, whereby the pitch and loudness of a vocal track can be used to synthesize "imposters" with different timbres, *e.g.,* a sine wave or flute, mixed with instrumental tracks, and used to measure false positives.

5. DISCUSSION

This work presents an approach to mining strongly labeled data from web-scale music collections for detecting vocal activity. Original recordings, containing vocals, are automatically paired with their instrumental counterparts, and differential information is used to estimate vocal activity over time. This signal can be used to train convolutional neural networks; the strongly labeled training data produces superior results to the weakly labeled setting, achieving state of the art performance. While analyzing errors made by our models, three distinct lessons stood out.

First, in addition to curation and mining, it is valuable to recall a third path to acquiring sufficiently large datasets: active learning. Imperfect models can be leveraged to make the annotation process more efficient by performing aspects of annotation that humans may find difficult or time-consuming, as well as prioritizing data as a function of model uncertainty. Here, for example, we observe that regions annotated as vocal tend to include brief pauses, no doubt resulting from the time and effort it would require to annotate at that level of detail. Alternatively, a performant model, like those described here, could segment audio into short, labeled excerpts for a human to verify or correct, eliminating a huge time cost. This would allow reliable data to be obtained at a faster rate.

Second, the application of machine learning to mined datasets can help identify particular challenges of a given task, unlocking new research directions. Here, our model identifies an interesting bias in the dataset that we had not previously considered, being the tight coupling between singing voice (timbre), melody (pitch), and production effects (loudness). Often in Western popular music, lead vocals carry the melody and tend to be one of the more prominent sources in the mix. Thus, in the dataset mined from a commercial music catalogue, instrumental versions not only lack vocal timbres, but prominent melodic contours are missing as well. This complex relationship is less obvious at a distance, but our experiments illustrate the challenges faced data-driven approaches to singing voice detection. By the same token, this also identifies an opportunity to build systems invariant to these dimensions.

Finally, these insights serve as a reminder that it is good practice to both design and evolve benchmarking datasets to encompass challenging test cases and known failure modes as they are identified. In our analysis, we find that the available benchmarking datasets consist mostly of musical content in which the melody is also voice, and therefore more "difficult" signals would help reliably discriminate between models. This content could be identified automatically via incremental evaluation methods, in which disagreement between machine estimations effectively prioritizes data to maximize discrimination between models [4].

5.1 Future Work

Perhaps the most logical next step for this work is to better augment training data, such that pitch and melodic information are well represented in negative examples. One possible approach, for example, would be to use the frequency of the vocal density estimate recovered in Section **??** to synthesize the melody with different timbres to be mixed into the instrumental recording. Whereas before entangled pairs contrast the presence of vocals, this approach would yield pairs that differ only in the timbre of the voice. Alternatively, additional sources could be leveraged for building models invariant to less relevant characteristics, such as instrumental content without a corresponding "original" version, or multitrack audio.

Additionally, more effort is required to advance evaluation methodology for automatic vocal activity detection. Multitrack datasets like MedleyDB, are a particularly promising route for rigorous benchmarking. The isolated vocal signal provides an optimal reference signal, while the other, non-vocal stems can be recombined as needed to deeply explore system behavior. We also recognize that larger, more diverse evaluation datasets are a prerequisite to advancing the state of the art in this domain. Thus, as a first step toward these ends, we provide machine estimations from our model over the datasets used here, as well as a large publicly available dataset (with audio) to facilitate the manual annotation process. [5] Though human effort is necessary to verify or correct these machine estimations, we share this data in the hope that it can serve as a starting point to accelerate the growth of labeled data for this task and facilitate efforts toward incremental evaluation.

[5] https://github.com/ejhumphrey/vox-detect-jams

6. REFERENCES

[1] Adam L Berenzweig and Daniel PW Ellis. Locating singing voice segments within music signals. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 119–122. IEEE, 2001.

[2] Rachel M Bittner, Justin Salamon, Mike Tierney, Matthias Mauch, Chris Cannam, and Juan Pablo Bello. Medleydb: A multitrack dataset for annotation-intensive MIR research. In *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, volume 14, pages 155–160, 2014.

[3] Judith C Brown. Calculation of a constant q spectral transform. *The Journal of the Acoustical Society of America*, 89(1):425–434, 1991.

[4] Ben Carterette and James Allan. Incremental test collections. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, pages 680–687. ACM, 2005.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pages 248–255. IEEE, 2009.

[6] Daniel PW Ellis, Brian Whitman, and Alastair Porter. Echoprint: An open music identification service. In *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR)*. ISMIR, 2011.

[7] Philippe Hamel, Simon Lemieux, Yoshua Bengio, and Douglas Eck. Temporal pooling and multiscale learning for automatic annotation and ranking of music audio. In *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR)*, pages 729–734, 2011.

[8] Eric J Humphrey and Juan Pablo Bello. Rethinking automatic chord recognition with convolutional neural networks. In *International Conference on Machine Learning and Applications (ICMLA)*, volume 2, pages 357–362. IEEE, 2012.

[9] Simon Leglaive, Romain Hennequin, and Roland Badeau. Singing voice detection with deep recurrent neural networks. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 121–125. IEEE, 2015.

[10] Bernhard Lehner, Gerhard Widmer, and Sebastian Bock. A low-latency, real-time-capable singing voice detection method with LSTM recurrent neural networks. In *Proceedings of the 23rd European Signal Processing Conference (EUSIPCO)*, pages 21–25. IEEE, 2015.

[11] Bernhard Lehner, Gerhard Widmer, and Reinhard Sonnleitner. On the reduction of false positives in singing voice detection. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7480–7484. IEEE, 2014.

[12] Matthias Mauch and Simon Dixon. pYIN: A fundamental frequency estimator using probabilistic threshold distributions. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 659–663. IEEE, 2014.

[13] Matthias Mauch, Hiromasa Fujihara, Kazuyoshi Yoshii, and Masataka Goto. Timbre and melody features for the recognition of vocal activity and instrumental solos in polyphonic music. In *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR)*, pages 233–238, 2011.

[14] Colin Raffel and Daniel PW Ellis. Large-scale content-based matching of MIDI and audio files. In *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR)*. ISMIR, 2015.

[15] Mathieu Ramona, Gaël Richard, and Bertrand David. Vocal detection in music with support vector machines. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1885–1888. IEEE, 2008.

[16] Lise Regnier and Geoffroy Peeters. Singing voice detection in music tracks using direct voice vibrato detection. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1685–1688. IEEE, 2009.

[17] Jan Schlüter. Learning to pinpoint singing voice from weakly labeled examples. In *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR)*, 2016.

[18] Jan Schlüter and Thomas Grill. Exploring data augmentation for improved singing voice detection with neural networks. In *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, pages 121–126, 2015.

[19] Andrew JR Simpson, Gerard Roma, and Mark D Plumbley. Deep Karaoke: Extracting vocals from musical mixtures using a convolutional deep neural network. In *Latent Variable Analysis and Signal Separation, International Conference on*, pages 429–436. Springer, 2015.

[20] D. Zwillinger and S. Kokoska, editors. *Probability and Statistics Tables and Formulae.* Chapman & Hall, New York, NY, 2000.