LYRICS TRANSCRIPTION FOR HUMANS: A READABILITY-AWARE BENCHMARK

Ondřej Cífka **Hendrik Schreiber** **Luke Miner** **Fabian-Robert Stöter**
AudioShake
ondrej@audioshake.ai, fabian@audioshake.ai

ABSTRACT

Writing down lyrics for human consumption involves not only accurately capturing word sequences, but also incorporating punctuation and formatting for clarity and to convey contextual information. This includes song structure, emotional emphasis, and contrast between lead and background vocals. While automatic lyrics transcription (ALT) systems have advanced beyond producing unstructured strings of words and are able to draw on wider context, ALT benchmarks have not kept pace and continue to focus exclusively on words. To address this gap, we introduce Jam-ALT, a comprehensive lyrics transcription benchmark. The benchmark features a complete revision of the JamendoLyrics dataset, in adherence to industry standards for lyrics transcription and formatting, along with evaluation metrics designed to capture and assess the lyric-specific nuances, laying the foundation for improving the readability of lyrics. We apply the benchmark to recent transcription systems and present additional error analysis, as well as an experimental comparison with a classical music dataset.

1. INTRODUCTION

Recent general-purpose automatic speech recognition (ASR) models trained on large datasets [1, 2] have shown a remarkable level of generalization, even improving the performance of automatic lyrics transcription (ALT) [3–5]. Remarkably, these state-of-the-art ASR models are able to take in larger temporal contexts and produce natural text with long-term coherence which, in the case of Whisper [2], includes punctuation and capitalization [6]. One may therefore ask how well these capabilities transfer from speech to lyrics. Moreover, producing a high-quality lyrics transcript suitable for user-facing music industry applications (e.g. to be displayed on streaming platforms or lyrics websites) presents some unique challenges, namely the need for specific formatting (e.g. line break placement, parentheses around background vocals) [7–9]. This calls for a new approach to ALT evaluation and development that accounts for these distinctive nuances.

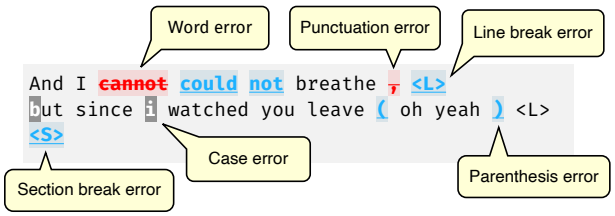


Figure 1: Error types captured by our metrics. Each token is classified as a word, punctuation mark, or parenthesis (enclosing background vocals). Special tokens are added in place of line and section breaks. Each token type is covered by a separate metric; differences in letter case are handled separately.

In ASR, the primary goal is a clear representation of what was said. To that end, formatting is helpful for improving the readability of transcripts [10]. Likewise, fillers like *um*, *uh*, *like*, and *you know* can be omitted to improve readability. Recent work [11] attempts to formalize this concern for clarity, proposing a novel metric geared towards assessing human readability. It employs human labelers, instructed to disregard filler words while, on the other hand, taking account of punctuation and capitalization errors that impact readability or alter the meaning of the text.

In music, on the other hand, lyrics are not simply a means of communicating meaning; they are a form of artistic expression, closely tied to the rhythm, melody, and emotionality of the song. For this reason, lyrics transcription requires a different set of considerations. Line breaks, often missing or arbitrarily placed in speech transcripts, are essential in lyrics for capturing rhyme, meter, and musical phrasing. Fillers like *oh yeah*, non-word sounds like *la-la-la* and contractions such as *I'ma* (vs. *I'm gonna*, *I am going to*) have prosodic significance, and their omission would disrupt the song's rhythm and rhyme scheme. Far from being an impediment to readability, they are key to any faithful rendition of a song for artist and fan alike.

We believe that readability-aware models for lyrics transcription have the potential to facilitate novel applications extending beyond the realms of metadata extraction and relatively crude karaoke subtitles. However, in order to advance in this research direction, the ability to accurately evaluate ALT systems in the aforementioned aspects is vi-

tal. To the best of our knowledge, existing ALT literature not only overlooks readability, but evaluates on datasets (e.g. [12–15]) that have not been designed specifically for ALT and lack some or all of the desirable features discussed above.

One of the datasets widely adopted by recent works [3, 4, 16–18] as an ALT test set is JamendoLyrics [14], originally a lyrics alignment benchmark. Its most recent ("MultiLang") version [19] contains four languages and a diverse set of genres, making it attractive as a testbed for lyrics-related tasks. However, we found that, in addition to lacking in the aspects discussed above, the lyrics are sometimes inaccurate or incomplete. While such lyrics may be perfectly acceptable as input for lyrics alignment (and indeed representative of a real-world scenario for that task), they are less suitable as a target for ALT.

To address these issues and help to guide future ALT research, we present the **Jam-ALT** benchmark, consisting of: (1) a revised version of JamendoLyrics MultiLang following a newly created annotation guide that unifies the music industry's conventions for lyrics transcription and formatting (in particular, regarding punctuation, line breaks, letter case, and non-word vocal sounds); (2) a comprehensive set of automated evaluation metrics designed to capture and distinguish different types of errors relevant to (1). The dataset and the implementation of the metrics are available via the project website. [1] Additionally, to explore the applicability of the proposed metrics to other datasets, we present results on the *Schubert Winterreise Dataset* (SWD) [20].

2. DATASET

Our first contribution is a revision of the JamendoLyrics MultiLang dataset [19] to make it more suitable as a lyrics transcription test set. Different sets of guidelines for lyrics transcription and formatting exist within the music industry; we consider guidelines by Apple [7], LyricFind [8], and Musixmatch [9], from which we extracted the following general rules:

1. Only transcribe words and vocal sounds audible in the recording; exclude credits, section labels, style markings, non-vocal sounds, etc.
2. Break lyrics up into lines and sections; separate sections by a single blank line.
3. Include each word, line and section as many times as heard. Do not use shorthands to indicate repetitions.
4. Start each line with a capital letter; respect standard capitalization rules for each language.
5. Respect standard punctuation rules, but never end a line with a comma or a period.
6. Use standard spelling, including standardized spelling for slang where appropriate.
7. Mark elisions (incomplete words) and contractions with an apostrophe.
8. Transcribe background vocals and non-word vocal sounds if they contribute to the content of the song.
9. Place background vocals in parentheses.

The original JamendoLyrics dataset adheres to rules 1, 3, and 8, partially 2 and 6 (up to some missing diacritics, misspellings, and misplaced line breaks), but lacks punctuation and is lowercase, thus ignoring rules 4, 5, 7, and 9. Moreover, as mentioned above, we found that the lyrics do not always accurately correspond to the audio.

To address these issues, we revised the lyrics in order for them to obey all of the above rules and to match the recordings as closely as possible. As the above rules are fairly unspecific, we created a detailed annotation guide where we have attempted to resolve minor discrepancies among the source guidelines [7–9] and fill in missing details (including language-specific nuances). This annotation guide is released together with the dataset.

Each lyric file was revised by a single annotator proficient in the language, then reviewed by two other annotators. In coordination with the authors of [19], one of the 20 French songs was removed following the detection of potentially harmful content.

Examples of lyrics before and after revision can be found on the project website.

3. METRICS

In this section, we first discuss our adaptation of the conventional *word error rate* (WER) metric and then our proposed precision and recall measures for punctuation and formatting. Our goal here is to design a comprehensive set of metrics that covers all possible transcription errors while allowing us to distinguish between different types of errors (see Fig. 1 for a visual overview of the error types). Note, however, that our goal is *not* to create metrics that completely align with the rules put forth in Section 2 or correlate with a specific notion of readability; the metrics should be general enough to apply to any plain-text lyrics dataset and adapt to its formatting style.

3.1 Word Error Rates

The standard speech recognition metric, WER, is defined as the edit distance (a.k.a. Levenshtein distance) between the *hypothesis* (predicted transcription) and the *reference* (ground-truth transcript), normalized by the length of the reference. If D, I, and S are the number of word *deletions*, *insertions*, and *substitutions* respectively, for the minimal sequence of edits needed to turn the reference into the hypothesis, and H is the number of unchanged words (*hits*), then:

$$\text{WER} = \frac{S + D + I}{S + D + H} = \frac{S + D + I}{N}, \tag{1}$$

where N is the total number of reference words.

Typically, the hypothesis and the reference are preprocessed to make the metric insensitive to variations in punctuation, letter case, and whitespace, but no single standard pre-processing procedure exists. In this work, we apply Moses-style [21] punctuation normalization and tokenization, then remove all non-word tokens. Before computing the WER, we lowercase each token to make the met-

[1] https://audioshake.github.io/jam-alt/

ric case-insensitive, but also keep track of the token's original form. To then measure the error in letter case, for every *hit* in the minimal edit sequence, we compare the original forms of the hypothesis and the reference token and count an error if they differ. We then compute a *case-sensitive word error rate* WER′ as:

$$\text{WER}' = \frac{S + D + I + E_{\text{case}}}{S + D + H} = \text{WER} + \frac{E_{\text{case}}}{N}, \quad (2)$$

where E_{case} is the number of casing errors. We include both variants (1) and (2) in our benchmark.

3.2 Punctuation and Line Breaks

Since the output of ASR systems traditionally lacks punctuation, a common ASR post-processing step – *punctuation restoration* [22] – consists of recovering it. This task is usually evaluated using precision and recall:

$$P = \frac{\text{\# correctly predicted symbols}}{\text{\# predicted symbols}}, \quad R = \frac{\text{\# correctly predicted symbols}}{\text{\# expected symbols}}. \quad (3)$$

In this original setting where the system only inserts punctuation and the words remain intact, computing the metrics is trivial. In contrast, in our end-to-end setting, the hypothesis and the reference may use different words, and hence computing the numerator in Eq. (3) requires an alignment between the two. We leverage the same alignment as used in Section 3.1, but computed on text that includes punctuation. Moreover, we extend this approach to account for line breaks, which, though traditionally ignored in speech data, are particularly important for lyrics.

We use the pre-processing from Section 3.1, but preserve punctuation tokens and, as in [23, 24], add special tokens in place of line and section breaks; this leaves us with five token types: word `W`, punctuation `P`, parenthesis `B` (separate due to its distinctive function), line break `L`, and section break `S`.[2] After computing the alignment between the hypothesis tokens and the reference tokens, we iterate through it in order to count, for each token type $T \in \{\texttt{W}, \texttt{P}, \texttt{B}, \texttt{L}, \texttt{S}\}$, its number of deletions D_T, insertions I_T, substitutions S_T, and hits H_T. In general, each edit operation is simply attributed to the type of the token affected (e.g. the insertion of a punctuation mark counts towards $I_{\texttt{P}}$). However, a substitution of a token of type T by a token of type $T' \neq T$ is counted as two operations: a deletion of type T (counting towards D_T) and an insertion of type T' (counting towards $I_{T'}$).

We can now use these counts to define a precision, recall, and F-1 metric for each token type:

$$P_T = \frac{H_T}{H_T + S_T + I_T}, \quad R_T = \frac{H_T}{H_T + S_T + D_T}, \quad F_T = \frac{2}{P_T^{-1} + R_T^{-1}}. \quad (4)$$

[2] We define a section break as one or more blank lines. Hence, every section break is explicitly preceded by a line break in our representation.

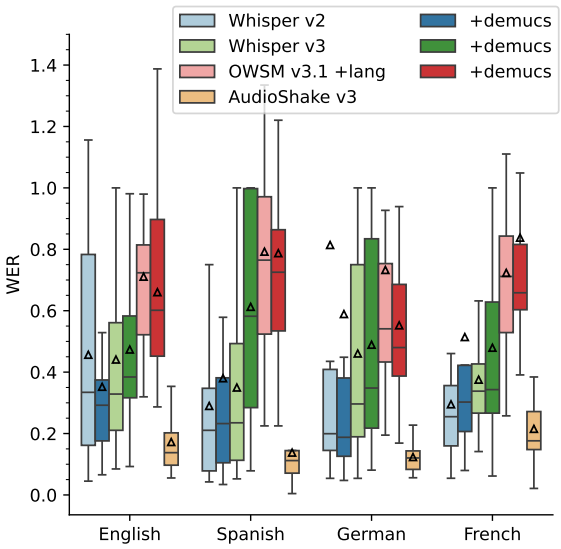


Figure 2: Song-level word error rates by language. Note that strong outliers occur; for clarity, they are not displayed here, but affect the means, which are indicated by triangles.

4. RESULTS

4.1 Benchmark Results

Table 1 shows the performance of various transcription systems on our benchmark. Fig. 2 shows the distributions of song-level word error rates by language.

We include two recent, freely available models capable of transcribing long, unsegmented audio: Whisper [2] (`large-v2` and `large-v3`) and OWSM 3.1 [25] (`owsm_v3.1_ebf`). For both models, we use Whisper-style long-form transcription with a beam size of 5. Both models have language identification capabilities, but may perform better if the correct language is specified; for Whisper, we evaluate both options, while for OWSM, for simplicity, we only evaluate with the language provided. For Whisper, which exhibits great variation between runs due to its stochastic decoding strategy, we report averages over 5 runs. We optionally use HTDemucs [26] to isolate the vocals from the input audio.

Whisper and OWSM are general-purpose speech recognition models and are not designed for lyrics transcription. To make a fairer comparison, we apply simple post-processing to their outputs to improve the formatting: (1) The models do not produce line breaks, but split their output into timestamped segments; we insert line breaks between these segments. (2) We remove unwanted end-of-line punctuation (all non-word characters except for `!?'"»)`) and uppercase the first letter of every line.[3]

We also evaluate LyricWhiz [4], a lyrics transcription system combining Whisper with the commercially available instruction-following language model ChatGPT [27]. We report averages over two outputs per song (English only), kindly provided by the LyricWhiz authors. Finally,

[3] Although we observed that this transformation tends to improve the outputs for Whisper and OWSM, in general, it may make evaluation results worse if the line break predictions are incorrect. For this reason, we do not include this step as a fixed part of our benchmark.

	All languages						English						Spanish		German		French	
	WER	WER′	F_{P}	F_{B}	F_{L}	F_{S}	WER	WER′	F_{P}	F_{B}	F_{L}	F_{S}	WER	WER′	WER	WER′	WER	WER′
Whisper v2	37.8	42.1	44.2	—	69.3	3.3	43.8	47.5	31.5	—	63.0	11.2	25.8	31.5	54.5	59.3	27.7	31.1
+lang	**27.9**	**32.6**	**45.0**	—	70.4	**3.7**	39.7	43.7	34.9	—	65.5	**11.6**	**21.9**	**27.7**	**19.9**	**26.0**	**27.1**	**30.5**
+demucs	44.5	49.8	41.6	—	61.2	—	**33.3**	**39.1**	**42.2**	—	53.9	—	39.6	46.5	65.2	70.4	43.3	46.9
+lang	33.5	39.3	39.4	—	60.6	—	35.6	41.3	41.8	—	53.4	—	34.9	42.2	23.9	30.4	38.2	42.1
Whisper v3	35.5	39.7	43.0	—	73.5	1.0	37.7	42.5	41.4	—	71.5	2.6	28.6	33.6	40.7	44.6	34.7	38.0
+lang	32.6	37.2	43.7	—	**73.9**	0.6	36.4	41.4	41.8	—	**72.5**	2.6	22.4	28.0	35.9	40.4	34.7	38.0
+demucs	48.0	51.6	33.0	—	65.7	—	43.0	47.2	25.8	—	66.9	—	61.5	64.9	43.5	47.4	44.9	48.2
+lang	46.6	50.4	33.7	—	65.8	—	43.0	47.2	25.8	—	66.9	—	58.6	62.1	40.8	44.9	44.9	48.3
OWSM v3.1+lang	69.3	75.0	22.5	**0.6**	37.8	—	68.6	74.0	22.3	—	42.7	—	73.3	78.5	63.3	71.8	71.6	75.7
+demucs	66.5	72.6	20.0	0.0	41.1	—	63.4	69.4	21.5	**0.0**	47.3	—	70.8	76.0	51.8	62.0	78.5	82.1
LyricWhiz	—	—	—	—	—	—	24.6	28.0	34.0	—	74.0	1.4	—	—	—	—	—	—
AudioShake v3	16.1	20.1	57.0	29.4	84.4	73.9	17.3	20.9	65.3	37.9	84.3	84.8	12.6	17.7	12.6	17.5	20.8	23.5
JamendoLyrics	11.1	29.6	—	—	93.3	85.3	14.4	29.6	—	—	88.1	77.9	14.0	29.1	5.0	37.6	10.3	23.3

Table 1: Benchmark results (all metrics shown as percentages). WER is word error rate, WER′ is case-sensitive WER, the rest are F-measures. +demucs indicates vocal separation using HTDemucs; +lang indicates that the language of each song was provided to the model instead of relying on auto-detection. Whisper results are averages over 5 runs with different random seeds, LyricWhiz over 2 runs; OWSM and AudioShake are deterministic, hence the results are from a single run. The best results achieved by open-source systems are shown in **bold**. LyricWhiz and AudioShake are listed separately, because they rely on proprietary technology. The last row shows metrics computed between the original JamendoLyrics dataset as the hypotheses and our revision as the reference. For full results by language, see the project website.

	All			EN	ES	DE	FR
	WER	F_{L}	F_{S}	WER			
Whisper v2	39.1	70.0	**2.8**	43.0	31.7	54.7	28.0
+lang	**28.8**	71.0	2.6	38.8	**27.9**	**19.8**	**27.4**
+demucs	46.2	61.5	—	**33.6**	43.9	65.5	44.1
+lang	34.8	61.2	—	36.1	39.3	23.9	38.9
Whisper v3	37.7	71.6	1.0	39.3	34.5	40.8	36.1
+lang	34.9	**72.3**	0.6	38.0	28.9	36.0	36.1
+demucs	49.6	65.3	—	44.3	65.8	43.5	45.7
+lang	48.3	65.4	—	44.3	63.1	40.8	45.7
OWSM v3.1+lang	70.3	39.0	—	69.9	75.7	63.5	71.9
+demucs	67.5	41.6	—	65.0	72.7	51.7	79.1
LyricWhiz	—	—	—	23.7	—	—	—
AudioShake v3	19.4	82.3	64.5	22.5	18.7	13.8	21.7
Jam-ALT	11.5	94.0	85.1	15.7	14.4	5.0	10.4

Table 2: Results with the original JamendoLyrics (i.e. before revision) as reference. The last row corresponds to our revision. See also the caption of Table 1.

as an example of an ALT system built with formatting and readability in mind, we include our in-house lyrics transcription system, which integrates vocal separation.

As a first general observation, consistent with previous studies [4, 5], the performance of Whisper models is relatively good, considering that they were not specifically designed for lyrics transcription. Among the formatting metrics, we highlight a high accuracy in line break prediction. This shows that, although the segments output by Whisper do not always impose a meaningful structure, in music, they do in many cases coincide with lyric lines.

Somewhat counter-intuitively, for Whisper, inputting isolated vocals (+demucs) tends to substantially degrade the results (with the single exception of large-v2 for English). Whisper's language identification mechanism also turns out to have a significant effect, in that disabling it and instead inputting the known language of the song (+lang) tends to result in a sizeable drop in WER, especially on languages different from English. This suggests that the language detected by Whisper is often incorrect.

We also observe that Whisper v3 does not necessarily perform better on lyrics than v2. In fact, the WER increases from 27.9 to 32.6 when comparing Whisper v2 +lang to v3 +lang.

The improvement of LyricWhiz over plain Whisper in terms of WER is clear and even sharper than reported in [4]. We also see some improvement in terms of line breaks and punctuation.

Regarding OWSM, its performance is far behind Whisper, with differences far larger than reported in [25] for speech, strongly suggesting that OWSM is poorly suited for ALT, at least without finetuning. With isolated vocals as input, the error is slightly reduced, but still large.

As for our own system, it outperforms all of the above on all metrics shown in Table 1, by a large margin, e.g. with a 57 % reduction in overall WER compared to Whisper v2. It is also the only one achieving acceptable accuracy for parentheses (B) and section breaks (S).

4.2 Effect of Revisions

The revisions described in Section 2 have enabled us to compute metrics related to letter case and punctuation, features that are missing from the original dataset. However, the revisions also involved correcting words and line breaks; to measure the effect of these corrections, we present in Table 2 the relevant metrics computed on the original JamendoLyrics data. Comparing Tables 1 and 2, we note that the revisions have mostly improved the results, notably reducing the overall WER (by 1.7, or 5.3 %, on average) for all systems, with Spanish seeing the sharpest drop (4.7, or 17.4 %, on average, likely due to fre-

quently missing accents in the original data). The general trends – in particular, the ranking based on WER and F_{L} – remain mostly unchanged.

To quantify the extent of our revisions more directly, we also evaluate both versions of the lyrics against each other and include the results as the last row in Tables 1 and 2. Remarkably, in terms of word tokens, Jam-ALT differs from JamendoLyrics by about 11 % (around 15 % for English and Spanish), which is substantially more than the difference between system performance on the two dataset versions. One potential explanation is that a significant number of the corrections correspond to low-intelligibility singing, which is prone to transcription errors, or to background vocals, which are susceptible to being omitted by transcription systems.

4.3 Error Analysis

In this section, we further analyze the errors made by selected systems on our benchmark.

First, we visualize in Fig. 3 how each type of edit operation contributes to the WER. Besides the basic edit operations (hits, substitutions, insertions, deletions), we include *case errors* from Section 3.1; that is, a hit with a difference in letter case is shown as a case error instead. Moreover, to account for small spelling differences, we consider a substitution as a *near hit* when the replacement differs from the reference in at most two letters. [4]

With Whisper, we observe that inputting separated vocals causes more insertions (and longer output) in v2, but more deletions (and shorter output) in v3. Upon inspecting the outputs, we find that Whisper has a general tendency to omit parts of the lyrics (often the entire song) and instead produce generic or irrelevant text, and that this is more frequent with separated vocals, especially with v3. On the other hand, OWSM shows a slight improvement with separated vocals, but its predictions contain significantly more substitutions, suggesting that they are more often incorrect on a word-by-word basis.

Next, we focus on errors in punctuation and formatting and investigate how often different token types are substituted for each other. To this end, we count the edit operations as in Section 3.2, but preserve the information about substitutions across the four non-word token types (`P`, `B`, `L`, `S`). We then present this information in a form akin to a *confusion matrix*, adding a special "null" token type ∅ to account for insertions and deletions.

The result is shown in Fig. 4 for three selected systems. Most errors are insertions and deletions, but another frequent type of error is the replacement of a line break by a punctuation mark, especially in Whisper models. This is explained by the fact that our guidelines forbid most end-of-line punctuation, and hence, when transcription omits a line break, inserting a punctuation mark in its place is often needed to maintain grammatical correctness.

[4] More precisely, we count a *near hit* if, after removing apostrophes from the two words, their character-level Levenshtein distance is at most 2, and strictly less than half the length of the longer of the two words. Examples include *an*/*and*, *gon'*/*gonna*, *there*/*their*/*they*/*them*, but not *a*/*an* or *this*/*that*.

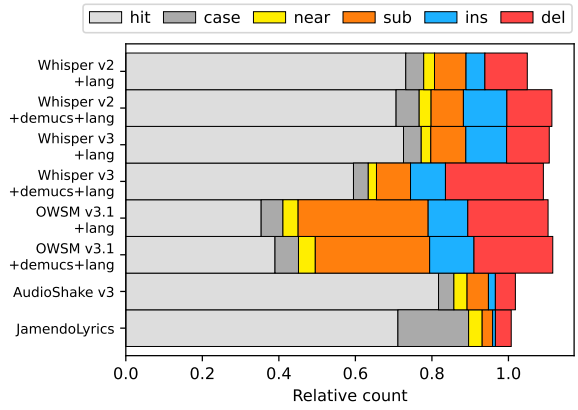


Figure 3: Word edit operation frequencies on our benchmark (one run per system). *Near* are substitutions that differ in few characters, *sub* are the remaining substitutions. *case* are hits with case errors, *hit* are the remaining (case-sensitive) hits. The rest are *insertions* and *deletions*. The frequencies are normalized by the reference length, so that:

- *hit* + *case* + *near* + *sub* + *del* = 1,
- WER = *near* + *sub* + *ins* + *del*,
- WER′ − WER = *case*,
- *hit*+*case*+*near*+*sub*+*ins* corresponds to the length of the prediction.

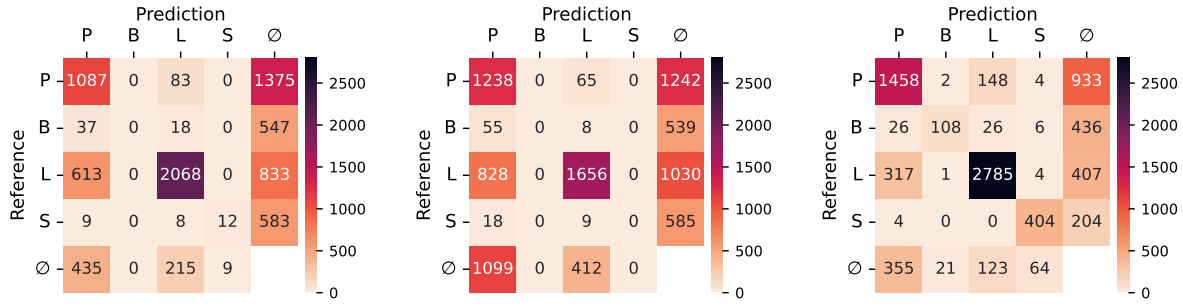
By manual inspection of the transcriptions, we find that Whisper tends to produce much longer lines than in the reference and frequently outputs periods (forbidden by our annotation guide as a sentence separator) and, occasionally, spuriously repeated punctuation.

4.4 Schubert Winterreise Dataset

To explore the application of the proposed metrics to other datasets, we additionally perform an evaluation on the *Schubert Winterreise Dataset* (SWD) [20]. SWD comprises nine audio versions of Franz Schubert's 24-song cycle *Winterreise*, along with symbolic representations, lyrics, and other annotations. An example of Romantic music based on early 19th century German poetry, it contrasts with JamendoLyrics and presents an interesting challenge for ALT. For our evaluation, we pick a single version, `SC06` (a 2006 live recording of singer Randall Scarlata), one of the two with audio publicly available.

The lyrics in SWD are formatted as poems – containing line and section breaks –, but their spelling and punctuation, mirroring an 1827 edition of the score [28], does not exactly match our annotation guide. To make them adhere to our punctuation and capitalization rules, we apply a simple transformation to the lyrics: replace all unwanted punctuation (`.;:`−) with commas, then remove all end-of-line commas and uppercase the first letter of each line. Note, however, that even after this transformation, the lyrics' obsolete spelling – predating the 1996 German orthography reform – violates our annotation guide to some extent (mainly in the usage of the letter *ß* and the treatment of elisions), which is expected to distort the WER.

We evaluate all models with the language provided (i.e.



(a) Whisper v2 `+lang` (b) Whisper v2 `+demucs+lang` (c) AudioShake v3

Figure 4: Edit operation counts on non-word (punctuation and formatting) tokens by token type (`P` = punctuation, `B` = parenthesis, `L` = line break, `S` = section break). ∅ denotes the absence of a token, i.e. it stands for insertion (on the *reference* axis) or deletion (on the *prediction* axis). Substitution of/by a *word* token is counted as an insertion/deletion, respectively. Only a single run per system is considered.

	WER	WER′	F_{P}	F_{L}	F_{S}
Whisper v2	**34.5**	**40.4**	**42.6**	**66.2**	—
`+demucs`	41.4	47.2	38.0	61.4	—
Whisper v3	59.0	63.8	40.0	63.6	—
`+demucs`	52.3	58.6	34.7	63.3	0.0
OWSM v3.1	75.6	82.5	12.9	39.6	**4.9**
`+demucs`	82.9	91.8	17.0	39.2	—
AudioShake v3	24.3	29.1	50.9	80.0	72.0

Table 3: Results on performance `SC06` from SWD. Only punctuation (`P`), line breaks (`L`) and section breaks (`S`) are included, as the ground truth lyrics do not contain any parentheses. Whisper results are averages over 5 runs with different random seeds. The best result in each column, excluding AudioShake, is shown in **bold**. For full results, see the project website.

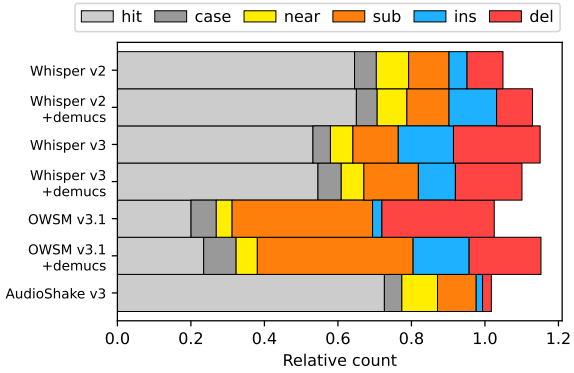


Figure 5: Word edit operation frequencies on SWD. See the caption of Fig. 3.

disabling language identification). The results are shown in Table 3 and further error analysis in Fig. 5. We notice substantially worse performance on SWD than the German section of our benchmark (Table 1): for example, WER for Whisper v2 `+lang` increased from 19.9 to 34.5. This likely reflects the more challenging nature of the dataset, but also possibly the mismatched spelling, as suggested by a higher frequency of near hits (see Fig. 5) than seen in Section 4.3 (Fig. 3).

5. DISCUSSION

Given our focus on formatting and punctuation, the question arises to what extent they are in fact dependent on the audio. In particular, could line and section boundaries be accurately predicted just from the textual context, e.g. based on metrical patterns, rhyme, syntax, and semantics? To answer this, we suggest an experiment where a human annotator is tasked with formatting given lyrics first without and then with access to the audio. Such a task would, however, be highly time-consuming and require expert annotators unfamiliar with the songs. As a proxy, one might instead train a *formatting restoration* model on lyrics or use a general-purpose instruction-following language model. Our attempts in this regard have only had limited success and we therefore leave such experiments for future work.

Another issue is that there may not always be a single correct division into lines and sections. For example, in a song with relatively short lines, it may be acceptable to join pairs of adjacent lines, especially in the absence of rhyme. Likewise, 4-line sections may be joined to create 8-line sections and so forth. However, it is not obvious how to relax the metrics to allow for this kind of variation. Doing so rigorously would likely require additional annotations, which is contrary to our goal of creating a set of generally applicable metrics. A possible solution compatible with this idea is to create multiple references and pick the best-scoring one during evaluation.

6. CONCLUSION

We have proposed Jam-ALT, a new benchmark for ALT, based on the music industry's lyrics guidelines. Our results show how existing systems differ in their performance on different aspects of the task, and we hope that the benchmark will be beneficial in guiding future ALT research.

7. ACKNOWLEDGMENT

We would like to thank Laura Ibáñez, Pamela Ode, Mathieu Fontaine, Claudia Faller, Constantinos Dimitriou,

and Kateřina Apolínová for their help with data annotation. We are also thankful to Meinard Müller and Hans-Ulrich Berendes for their helpful comments on the manuscript.

8. REFERENCES

[1] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems*, 2020. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/92d1e1eb1cd6f9fba3227870bb6d7f07-Abstract.html

[2] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. Mcleavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *Proceedings of the 40th International Conference on Machine Learning*, vol. 202. PMLR, 23–29 Jul 2023, pp. 28 492–28 518. [Online]. Available: https://proceedings.mlr.press/v202/radford23a.html

[3] L. Ou, X. Gu, and Y. Wang, "Transfer learning of wav2vec 2.0 for automatic lyric transcription," in *Proceedings of the 23rd International Society for Music Information Retrieval Conference (ISMIR 2022)*, Bengaluru, India, 2022, pp. 891–899.

[4] L. Zhuo, R. Yuan, J. Pan, Y. Ma, Y. Li, G. Zhang, S. Liu, R. B. Dannenberg, J. Fu, C. Lin, E. Benetos, W. Chen, W. Xue, and Y. Guo, "LyricWhiz: Robust multilingual zero-shot lyrics transcription by whispering to ChatGPT," in *Proceedings of the 24th International Society for Music Information Retrieval Conference (ISMIR 2023)*, Milan, Italy, 2023.

[5] J. Wang, C. Leong, Y. Lin, L. Su, and J. R. Jang, "Adapting pretrained speech model for Mandarin lyrics transcription and alignment," in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU 2023)*. IEEE, 2023, pp. 1–8. [Online]. Available: https://doi.org/10.1109/ASRU57964.2023.10389800

[6] L. R. S. Gris, R. Marcacini, A. C. Júnior, E. Casanova, A. da Silva Soares, and S. M. Aluísio, "Evaluating OpenAI's Whisper ASR for punctuation prediction and topic modeling of life histories of the Museum of the Person," *CoRR*, vol. abs/2305.14580, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2305.14580

[7] Apple, "Review guidelines for submitting lyrics," 2023, accessed: 2023-09-18. [Online]. Available: https://web.archive.org/web/20230718032545/https://artists.apple.com/support/1111-lyrics-guidelines

[8] LyricFind, "Lyric formatting guidelines," 2023, accessed: 2023-09-18. [Online]. Available: https://web.archive.org/web/20230521044423/https://docs.lyricfind.com/LyricFind_LyricFormattingGuidelines.pdf

[9] Musixmatch, "Guidelines," 2023, accessed: 2023-09-23. [Online]. Available: https://web.archive.org/web/20230920234602/https://community.musixmatch.com/guidelines

[10] D. A. Jones, F. Wolf, E. Gibson, E. Williams, E. Fedorenko, D. A. Reynolds, and M. Zissman, "Measuring the readability of automatic speech-to-text transcripts," in *Proc. 8th European Conference on Speech Communication and Technology (Eurospeech 2003)*, 2003, pp. 1585–1588.

[11] Apple, "Humanizing word error rate for ASR transcript readability and accessibility," 2024, accessed: 2024-04-09. [Online]. Available: https://machinelearning.apple.com/research/humanizing-wer

[12] C.-L. Hsu and J.-S. R. Jang, "On the improvement of singing voice separation for monaural recordings using the MIR-1K dataset," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 2, pp. 310–319, 2010.

[13] 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 *Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR 2018)*. ISMIR, Nov. 2018, pp. 431–437. [Online]. Available: https://doi.org/10.5281/zenodo.1492443

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

[15] Y. Wang, X. Wang, P. Zhu, J. Wu, H. Li, H. Xue, Y. Zhang, L. Xie, and M. Bi, "Opencpop: A high-quality open source Chinese popular song corpus for singing voice synthesis," in *Interspeech 2022, 23rd Annual Conference of the International Speech Communication Association, Incheon, Korea, 18-22 September 2022*, H. Ko and J. H. L. Hansen, Eds. ISCA, 2022, pp. 4242–4246. [Online]. Available: https://doi.org/10.21437/Interspeech.2022-48

[16] C. Gupta, E. Yilmaz, and H. Li, "Automatic lyrics alignment and transcription in polyphonic music: Does background music help?" in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 496–500. [Online]. Available: https://doi.org/10.1109/ICASSP40776.2020.9054567

[17] E. Demirel, S. Ahlbäck, and S. Dixon, "MSTRE-Net: Multistreaming acoustic modeling for automatic lyrics transcription," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference (ISMIR 2021)*, J. H. Lee, A. Lerch,

Z. Duan, J. Nam, P. Rao, P. van Kranenburg, and A. Srinivasamurthy, Eds., 2021, pp. 151–158. [Online]. Available: https://archives.ismir.net/ismir2021/paper/000018.pdf

[18] E. Demirel, S. Ahlbäck, and S. Dixon, "Low resource audio-to-lyrics alignment from polyphonic music recordings," in *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 586–590. [Online]. Available: https://doi.org/10.1109/ICASSP39728.2021.9414395

[19] S. Durand, D. Stoller, and S. Ewert, "Contrastive learning-based audio to lyrics alignment for multiple languages," in *2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Rhodes Island, Greece, 2023, pp. 1–5.

[20] C. Weiß, F. Zalkow, V. Arifi-Müller, M. Müller, H. V. Koops, A. Volk, and H. G. Grohganz, "Schubert winterreise dataset: A multimodal scenario for music analysis," *ACM Journal on Computing and Cultural Heritage*, vol. 14, no. 2, pp. 25:1–25:18, 2021. [Online]. Available: https://doi.org/10.1145/3429743

[21] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open source toolkit for statistical machine translation," in *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*. Prague, Czech Republic: Association for Computational Linguistics, Jun. 2007, pp. 177–180. [Online]. Available: https://aclanthology.org/P07-2045

[22] V. F. Pais and D. Tufis, "Capitalization and punctuation restoration: a survey," *Artificial Intelligence Review*, vol. 55, no. 3, pp. 1681–1722, 2022. [Online]. Available: https://doi.org/10.1007/s10462-021-10051-x

[23] E. Matusov, P. Wilken, and Y. Georgakopoulou, "Customizing neural machine translation for subtitling," in *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 82–93. [Online]. Available: https://aclanthology.org/W19-5209

[24] A. Karakanta, M. Negri, and M. Turchi, "Is 42 the answer to everything in subtitling-oriented speech translation?" in *Proceedings of the 17th International Conference on Spoken Language Translation*. Online: Association for Computational Linguistics, Jul. 2020, pp. 209–219. [Online]. Available: https://aclanthology.org/2020.iwslt-1.26

[25] Y. Peng, J. Tian, W. Chen, S. Arora, B. Yan, Y. Sudo, M. Shakeel, K. Choi, J. Shi, X. Chang, J. Jung, and S. Watanabe, "OWSM v3.1: Better and faster open Whisper-style speech models based on E-Branchformer," *CoRR*, vol. abs/2401.16658, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2401.16658

[26] S. Rouard, F. Massa, and A. Défossez, "Hybrid Transformers for music source separation," in *2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Rhodes Island, Greece, 2023, pp. 1–5.

[27] OpenAI, "Introducing ChatGPT," OpenAI Blog. [Online]. Available: https://openai.com/blog/chatgpt

[28] F. Schubert, "Winterreise. Ein Cyclus von Liedern von Wilhelm Müller," Gesänge für eine Singstimme mit Klavierbegleitung, Edition Peters, No.20a, n.d. Plate 9023, 1827. [Online]. Available: http://ks4.imslp.info/files/imglnks/usimg/9/92/IMSLP00414-Schubert_-_Winterreise.pdf