LLM-Generated Song Lessons: Perceived Usefulness and Shared-Cache Cost Analysis

Georgi Trajkov
geotrajkov0@gmail.com
Jožef Stefan Institute
Jožef Stefan International Postgraduate School
Ljubljana, Slovenia

Abstract

LangSong is a mobile-first prototype that turns supplied song lyrics into a language lesson using a schema-constrained large language model (LLM), with analyses and exercises cached across users. We examine perceived feature usefulness, logged use and modelled cost per user. In a self-guided convenience study with 16 participants, tap-to-gloss and line translation received useful ratings from 11/16 and 10/16 respondents; transliteration, shown only for non-Latin scripts, was rated 3 to 5 by the four who could see it. Free exploration after the tour lasted a median of 0.8 minutes, and 11 ratings referred to features the participant had never opened. Per-call costs measured through the API (918 calls over three songs) feed a cost model: with the shared store, the model cost per user at 15 songs a month falls from $0.1406 at 100 users to $0.0211 at a million ($0.2034 without sharing), and lessons for the whole Italian catalogue of 2,500,000 songs are a one-off of about $30,625, a year of 851 subscribers. These are scenarios, not measured operating costs, and cover inference only; lyrics licensing and streaming-service terms were not assessed and may prove the harder constraints.

Keywords

computer-assisted language learning, large language models, song lyrics, perceived usefulness, caching, inference cost

1 Introduction

Songs help with learning a language. When listening to a song in the language being learned, each replay is an unplanned rehearsal [1], and repeated listening builds vocabulary without deliberate study [2, 3]. Following such a song is harder than it should be: a listener who wants to understand, for example, a Macedonian song switches between a streaming service, a lyrics site, a transliteration if they do not know the script, and a published translation if one exists or a translation tool such as a Large Language Model (LLM). Apps for learning languages through songs exist, but they have several limitations (Section 2).

We developed LangSong to cover all of this ground and more, as a novel approach that relies on what LLMs now make feasible. One schema-constrained LLM call turns any supplied lyrics into a lesson: transliteration, line by line translation, per-word glosses and grammar notes. From the lesson the learner can open a deeper explanation of the grammar of any word, ask an AI tutor questions, practise generated exercises, or just follow the real time lyrics while the song plays in the app. No lesson is written by hand, so any song can be covered, instead every lesson, Deeper card and exercise set comes from an LLM call. We evaluated the usefulness of this approach via user testing and survey, while also asking how the number of users affects the marginal cost of analysis. Output quality and rights to the source lyrics are separate questions, taken up in Section 6.

We ask two research questions. **RQ1** (value): which features do learners rate most useful? **RQ2** (cost): what is the marginal LLM cost of each feature, and how does a shared store of finished analyses change cost per user as the number of users grows?

2 Background and Related Work

2.1 Music, LLMs and language learning

Singing new phrases aids recall more than speaking them [2], and repeated listening yields incidental vocabulary gains [3]. Apps like LingoClip (formerly LyricsTraining) offer gap-fill, karaoke and a word translator over a selected catalogue, not lessons for any song in a streaming catalogue such as Spotify's [4, 5]. Duolingo's paid tier adds AI conversation practice [6] but no song-based course.

LLMs can explain word meanings in context, produce grammar notes on demand and generate exercises [7, 8]; generated exercises have been rated close to human-written ones [9]. Generative-AI research in language learning is concentrated on learners of English and on chatbots [10]; song-based, multi-language use is largely unexplored.

2.2 Popularity skew

LangSong stores each finished analysis and serves it to every later listener, so whether that saves cost depends on how concentrated demand is: web requests follow Zipf-like distributions, where the i-th most popular item is requested about $1/i$ as often as the first, and cache performance depends on this skew and on catalogue size [11]. Music demand is similarly skewed [12, 13], and the long tail is where "any song" matters [14].

3 The LangSong Prototype

3.1 Features

LangSong is built as a web app requiring three main components: a streaming service (Spotify), lyric data (Musixmatch) and an LLM provider (OpenAI). The song is in the language being learned; the language the app translates and explains into, which we call the target language, is currently English only. It syncs with the user's streaming library, and with one click any of their songs can be translated. For each song one LLM call returns a lesson: line-by-line transliteration in a per-language Latin standard, line-by-line translation, for every word its dictionary form, part of speech and short meaning (a gloss), and four to six grammar notes. Once in the lesson, the user can listen as in any streaming service while the current line is highlighted with its transliteration and

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
Information Society 2026, Ljubljana, Slovenia
© 2026 Copyright held by the owner/author(s).
https://doi.org/10.70314/is.2026.sikdd.111

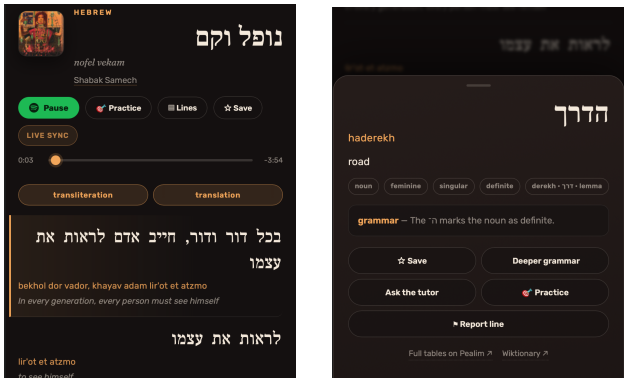

Figure 1: The lesson view during playback, with the current line highlighted and its transliteration and translation (left), and a word card with the gloss, grammar tags and the Deeper, tutor and practice buttons (right).

translation (Figure 1; timed lyrics were available for 87.8% of the songs with lyrics in the prototype's catalogue). Two song-level exercises follow: the Lines trainer plays one line at a time and asks the learner to fill or spell a missing word, and Practice offers ten generated exercises per song: word meanings in both directions, a lyric line with a blank, questions on the grammar notes, the vocabulary in new sentences, and two typed answers in the original script. Tapping any word opens its card: dictionary form, part of speech (noun, verb, …), meaning and a one-line grammar note, with buttons to save the word, open a Deeper grammar explanation (the word's full morphology, forms and examples), ask the AI tutor (a chat that knows the word and its line in the song) or practise the word (exercises generated for that word).

3.2 One-call analysis and the shared cache

Lyrics are normalised and hashed, and the hash names the cached analysis, shared by all users; two catalogue entries with the same lyrics (a remaster, a re-recording) are therefore analysed once. Only 63.3% of the lines in our corpus are unique (4,141 of 6,540); the model annotates each unique line once and the app copies the result to its repeats, to save output tokens (the full lyrics still go in as context). The call uses OpenAI `gpt-5.6-luna` through the Codex CLI at low reasoning effort with JSON-schema output. A validator checks that every line came back with a Latin-only transliteration (if lyrics are in non-Latin script) and has correct structure; a follow-up call repairs lines if there are errors. Deeper cards and practice sets are stored by word and song and shared the same way; the song level practice sets are pre-generated right after analysis, so their cost is shared too. AI Tutor sessions are not cached.

The app runs on a single small virtual server. The snapshot used below holds the songs analysed so far by the connected users and the study: 148 songs in 14 languages, Hebrew the largest group, with 59 Deeper grammar cards.

4 User Study Results

4.1 Protocol and observation window

The self-guided study involved 16 participants on their own phones (age bands 18–24 through 55+; 14 native South Slavic speakers; 12 past Duolingo users). Participants received a personal link, chose a language and song, followed an in-app tour of the features, then explored before completing a questionnaire. They tested 7 languages; asked for their knowledge level in that language, 6 answered none or a few words, 6 A1–A2 and 4 B1–B2. The questionnaire asked participants to rate how useful each feature was for understanding the song (perceived usefulness [15]), from 1 to 5 or "didn't try"; to name the one feature they would keep and the one they trusted least; and to say what they would use LangSong for, in which form (standalone app, inside Spotify, browser extension), what they would pay per month and whether the translations looked accurate. All 18 exported responses include consent and distinct participant codes; logs were joined by normalised code. Two respondents whose native language was the language they tested are excluded, leaving 16. Two participants left the tour for the practice exercises before its last steps; for one the in-app browser also reloaded the page mid-tour.

Table 1: Optional features: used at any point in the study (one participant also on an earlier account) and rated useful (4 or 5) by those who rated it.

Feature	Used	Rated useful
Deeper grammar card	13	5/10
Tutor chat	6	1/5
Practice exercises	12	5/11

Logged use is reported alongside the ratings, because stated and revealed preference can diverge [16]. After the tour, the 14 participants with a completed tour explored for a median of 0.8 minutes (mean 1.8) before opening the questionnaire.

4.2 Ratings

Figure 2 shows the ratings. A rating counts only when the log shows the participant used the feature, in the study or on an earlier account: 11 answers rated a feature the participant never opened and are treated as "didn't try". Tap-to-gloss was rated 4 or 5 by 11/16 respondents, followed by translation (10/16), practice (5/11), Deeper (5/10) and tutor chat (1/5). Transliteration appears only for non-Latin scripts: the twelve Latin-script testers rated it anyway, those ratings are dropped, and the four Russian and Japanese testers rated it 5, 5, 4 and 3. Asked which single feature they would keep, 6 chose translation, 4 gloss, 3 practice, 2 Deeper, 1 transliteration and 0 the tutor; asked which they trusted least, most named the tutor (5; 6 named none).

4.3 Logged use

The tour makes every participant tap a word and lets them try or skip the other steps; Table 1 counts who used each optional feature at any point in the study. The tutor was used by 6 participants, of whom 1/5 rated it useful. Free exploration after the tour was brief (median 0.8 minutes), and not every rating reflects use, as the transliteration answers show; Figure 2 therefore shows the data after cleaning.

5 Cost and Scale

5.1 Measurement

To emulate what a production deployment would use, we processed 3 cached songs (Hebrew, Russian and German) through the OpenRouter API with `gpt-5.6-luna` and DeepSeek V4 Flash: the lesson (analysis, a repair when the validator asks for one, which happened once, on Luna; the practice set), then a Deeper card and a word-practice set for every unique word, and one tutor turn,

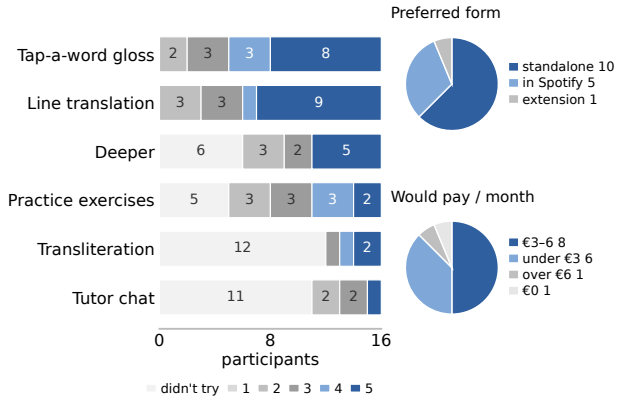

Figure 2: Questionnaire results after cleaning ($n = 16$): native-language testers excluded, transliteration counted only for non-Latin-script testers, features never opened counted as not tried.

Table 2: Measured API cost per call with Luna: means over 3 songs, n calls, input and output tokens, the share of input served from the prompt cache, and billed USD.

Feature	n	Input	Output	Cached	USD
Song analysis	3	2,029	7,904	0%	0.0100
Song practice	3	4,736	895	0%	0.0023
Deeper card	222	1,437	649	90%	0.0008
Word practice	222	1,190	600	59%	0.0008
Chat turn	3	599	85	0%	0.0002

919 billed calls in all ($0.52); one DeepSeek Deeper call failed validation ($0.02) and is left out of Table 2, which reports the cost per call. The prompts are the app's own, reordered so that the fixed rules and the lyrics form a shared prefix for provider prompt caching (the cached share in Table 2). OpenAI caches only prefixes of at least 1,024 tokens, so the calls made once per song, which share only their short instructions, show no cached input, while the word calls share the rules and the lyrics. DeepSeek ran with thinking off (with it on, one analysis cost as much as Luna's).

Costs are the billed amounts at the providers' rates [17]: Luna $0.20 per million input tokens, 1.25 times that for a prompt of at least 1,024 tokens being written to the cache, $0.02 when read from it and $1.20 per million output tokens; V4 Flash as hosted on OpenRouter $0.08, $0.016 and $0.16. Optional tools, fast processing and regional uplifts are outside the scenario.

5.2 Model

Let π_{in}, π_c and π_{out} be the prices per input, cached-input and output token, and t_{in}, t_c, t_{out} the input, cached and output tokens of one call of feature f (Table 2 uses the billed values directly):

$$C_f = \pi_{\text{in}}(t_{\text{in}} - t_c) + \pi_c t_c + \pi_{\text{out}} t_{\text{out}}. \tag{1}$$

Let songs be requested with Zipf probabilities $p_i = i^{-s}/H_{M,s}$ (the i-th most popular song is requested in proportion to i^{-s}), $H_{M,s} = \sum_{i=1}^{M} i^{-s}$, over a catalogue of M songs and each user make k independent song requests per month, with replacement; after N users the expected number of distinct songs analysed is

$$\mathbb{E}[D(N)] = \sum_{i=1}^{M} \left(1 - (1 - p_i)^{Nk}\right). \tag{2}$$

A new song costs at least its lesson, $C_{\text{analysis}} + C_{\text{prac}}$. Opening a word costs nothing, since its card comes from the analysis; asking for a word's Deeper card costs C_{deeper} and practising the word a word-practice set, C_{wordprac}. The two are requested independently, so the cost of one new song grows with the Deeper cards w_d and word-practice sets w_p generated for it, from the lesson alone ($w_d = w_p = 0$) to every unique word (Table 3):

$$C_{\text{new}} = C_{\text{analysis}} + C_{\text{prac}} + w_d\, C_{\text{deeper}} + w_p\, C_{\text{wordprac}}. \tag{3}$$

The baselines $w_d = 0.97$ and $w_p = 0.53$ are the Deeper cards and word-practice sets per song in production (33 Deeper cards and 18 sets over the 34 songs on which any word was opened). Repair calls (35 after 21 of the 83 logged analyses, some needing several rounds) cost $0.0015 each on Luna and are left out of C_{new}: at the logged rate they add under 5% to the bundle and can likely be eliminated with improvements to the system. With F the fixed monthly infrastructure cost and q AI tutor chat turns per user-month, the monthly cost per user with the shared cache, the same without any cache, and the cache hit rate are

$$c(N) = \frac{C_{\text{new}}\,\mathbb{E}[D(N)]}{N} + q\,C_{\text{chat}} + \frac{F}{N}, \tag{4}$$

$$c_{\text{nc}}(N) = k\,C_{\text{new}} + q\,C_{\text{chat}} + \frac{F}{N}, \tag{5}$$

$$h(N) = 1 - \mathbb{E}[D(N)]/(Nk). \tag{6}$$

The scenario sets $k = 15$ song requests per user-month, $q = 3$ chat turns, $s = 1$, $F = 0$ so that the curve is the model cost alone, and $M = 2,500,000$, an Italian-sized catalogue: Italian is 2–3% of the WASABI lyrics corpus [18, 19] and Spotify holds over 100 million tracks [20], while MusicBrainz lists 44,449 Italian-language and 112,702 Italian releases, on the order of 300,000 to 800,000 recordings [21]. τ = $0.10 per user-month, a thirtieth of the subscription below, is a reference point for comparing scenarios, not a target. Equation 2 models one month from an empty store; a persistent catalogue costs less in later months.

5.3 Scale

Without sharing, a user who requests 15 songs a month costs $0.2034 in model calls. With the shared store the cost per user goes from $0.1406 per month at 100 users to $0.0797 at 10,000 and $0.0211 at 1,000,000, as the hit rate rises from 31.0% to 89.9%, towards the AI tutor cost of $0.0007. Ten cents per user-month is reached at 2,152 users; the fall is slow because at 15 requests a month a large catalogue keeps being explored. With DeepSeek V4 Flash's measured costs the cost per user is $0.0334 at 100 users and $0.0050 at a million, under ten cents even without sharing ($0.0483).

Against a hypothetical subscription of $3 a month, a user costs more than they pay only through generation nobody has requested before: at $0.0148 per new song with 1.5 Deeper cards and as many word-practice sets, a user must add 203 new songs a month before their model cost passes the subscription, or 22 if every word of every song is generated (853 and 95 with Flash). A cap of 50 new analyses per user-month keeps new-song generation at that rate to $0.74 (25% of the subscription); word-level generation on songs already analysed and chat would need limits of their own. Such a cap is worth enforcing for niche languages, where few later listeners share the analysis; for popular pairs

Table 3: Generating ahead of demand: cost per song as a lesson only (analysis and practice) and fully pre-generated (plus a Deeper card and a practice set for each of the median 74.5 unique words); the one-off cost of an Italian-sized catalogue of 2,500,000 songs either way; and the number of subscribers at a hypothetical $3 a month whose first year covers it.

Model	Per song		Whole catalogue		Subscriber-years	
	Lesson	Full	Lessons	Full	Lessons	Full
Luna	$0.012	$0.14	$30,625	$345,204	851	9,589
DeepSeek V4 Flash	$0.003	$0.03	$7,352	$79,293	205	2,203

such as English speakers learning Italian, every new analysis populates the catalogue for other users, so heavy users are cheap content acquisition. Table 3 turns the same per-call costs into what generating a whole catalogue ahead of demand would cost.

6 Discussion and Limitations

Feature interpretation. Useful ratings for Deeper, practice and the tutor came after a short session that the tour had just walked through, and some were given without use. The result supports a simple annotated-song screen with optional learning controls; it does not show that learners reject those features.

What participants want. Respondents most often preferred a standalone app (10/16), followed by an in-Spotify feature (5) and a browser extension (1). Spotify and YouTube Music each received 11 playback-integration selections. AI-assisted playlists received 10 votes and lesson packs 9. 15/16 selected a nonzero hypothetical monthly price (not observed purchasing): 6 under €3, 8 at €3–6 and 1 above €6. Learning and song understanding received 12 and 12 selections, with 8 choosing both. Of 14 respondents able to judge perceived translation accuracy, 11 answered yes and 3 mostly; there was no independent linguistic evaluation.

Where scale helps. On demand, the modelled inference cost per user stays below fifteen cents a month even at a hundred users ($0.1406), so on modelled inference cost alone a hypothetical $3 subscription covers a language offered on demand from the start; the question is when generating ahead of demand pays off. Lessons for the whole Italian catalogue of 2,500,000 songs are a one-off of about $30,625 with Luna or $7,352 with DeepSeek V4 Flash (Table 3), the first year of 851 or 205 subscribers at $3 a month; scenario catalogues of 300,000 songs (about the Hebrew repertoire) and 100,000 (Macedonian) cost $3,675 and $1,225. Pre-generating a Deeper card and a practice set for each of the 74.5 unique words of every song raises the Italian figure to $345,204 (9,589 subscriber-years), worth doing only for the head of the catalogue. Each target language is a separate lesson (translation and glosses change), so language pairs multiply the catalogue cost; English-target pairs cover most learners, and niche pairs such as Hebrew songs for Macedonian speakers stay on demand.

Cost and limitations. Shared caching can reduce modelled inference cost, but the result depends on request concentration, catalogue size, word practice, chat demand and per-call costs measured on three songs through one API gateway, whose cached shares come from bulk generation with a warm prefix that sparse on-demand word calls would not reach. It excludes licensing, support, acquisition and infrastructure scaling. The cost analysis rests on one usage model, the same for every user: k independent requests a month from one Zipf catalogue at the production rates of Deeper cards and word practice. Real use could differ substantially; whether users keep to one language pair or several, and outlier users, are out of scope. Under these assumptions inference cost appears manageable for a wider deployment; lyrics licensing and streaming-service terms may matter more, but neither was assessed (no licensing quote or feasibility study). The study is a small convenience sample recruited from the author's network, using shared playback and self-reported proficiency. Device and playback failures can depress logged use; the counts support no causal or population-wide conclusions.

7 Conclusion

LangSong demonstrates shared generation and reuse of song-based language annotations. In a 16-person formative study, gloss and translation received the most useful ratings; free exploration after the tour was brief, and some ratings referred to features the participant had not opened. Cost per user falls with the number of users because every analysis is shared, from $0.1406 to $0.0211 per month between a hundred and a million users under the baseline assumptions, and lessons for a whole Italian-sized catalogue cost about a year of 851 subscribers. Under those assumptions inference cost appears manageable; lyrics licensing and streaming-service terms, not assessed here, may prove the harder constraints. Further work should assess them alongside usage, learning and capacity, and evaluate the lessons linguistically.

Acknowledgements

The research was supported through the European project Raido (GA 101135800). The author acknowledges usage of LLMs in grammar refinement and overall manuscript organisation and takes full responsibility for the final content of this publication.

References

[1] Tim Murphey. 1990. The song stuck in my head phenomenon: a melodic din in the LAD? *System*, 18, 1, 53–64. doi:10.1016/0346-251X(90)90028-4.

[2] Karen M. Ludke, Fernanda Ferreira, and Katie Overy. 2014. Singing can facilitate foreign language learning. *Memory & Cognition*, 42, 1, 41–52. doi:10.3758/s13421-013-0342-5.

[3] Niousha Pavia, Stuart Webb, and Farahnaz Faez. 2019. Incidental vocabulary learning through listening to songs. *Studies in Second Language Acquisition*, 41, 4, 745–768. doi:10.1017/S0272263119000020.

[4] LingoClip. 2025. LingoClip (formerly LyricsTraining): learn languages with music. https://lingoclip.com/. Accessed 2026-08-28; previously https://lyricstraining.com/. (2025).

[5] Brian S. McKinney. 2025. LyricsTraining and LingoClip. *CALICO Journal*, 42, 2, 336–344. doi:10.3138/calico-2024-0005.

[6] Duolingo. 2023. Introducing Duolingo Max, a learning experience powered by GPT-4. https://blog.duolingo.com/duolingo-max/. (Mar. 2023).

[7] Lucas Kohnke, Benjamin Luke Moorhouse, and Di Zou. 2023. ChatGPT for language teaching and learning. *RELC Journal*, 54, 2, 537–550. doi:10.1177/00336882231162868.

[8] Enkelejda Kasneci et al. 2023. ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and Individual Differences*, 103, 102274. doi:10.1016/j.lindif.2023.102274.

[9] Changrong Xiao, Sean Xin Xu, Kunpeng Zhang, Yufang Wang, and Lei Xia. 2023. Evaluating reading comprehension exercises generated by LLMs: a showcase of ChatGPT in education applications. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*. Association for Computational Linguistics, Toronto, Canada, 610–625. doi:10.18653/v1/2023.bea-1.52.

[10] Belle Li, Yaling Lily Tan, Chaoran Wang, and Victoria Lowell. 2025. Two years of innovation: a systematic review of empirical generative AI research in language learning and teaching. *Computers and Education: Artificial Intelligence*, 9, 100445. doi:10.1016/j.caeai.2025.100445.

[11] Lee Breslau, Pei Cao, Li Fan, Graham Phillips, and Scott Shenker. 1999. Web caching and Zipf-like distributions: evidence and implications. In *Proceedings of IEEE INFOCOM '99*. Vol. 1. IEEE, New York, NY, 126–134. doi:10.1109/INFCOM.1999.749260.

[12] Matthew J. Salganik, Peter Sheridan Dodds, and Duncan J. Watts. 2006. Experimental study of inequality and unpredictability in an artificial cultural market. *Science*, 311, 5762, 854–856. doi:10.1126/science.1121066.

[13] Peter L. Ormosi and Franco Mariuzzo. 2022. Independent vs major record labels: do they have the same streaming power (law)? *Economics Bulletin*, 42, 3, 1510–1526. https://www.accessecon.com/Pubs/EB/2022/Volume42/EB-22-V42-I3-P126.pdf.

[14] Chris Anderson. 2006. *The Long Tail: Why the Future of Business Is Selling Less of More*. Hyperion, New York. ISBN: 978-1-4013-0237-5.

[15] Fred D. Davis. 1989. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, 13, 3, 319–340. doi:10.2307/249008.

[16] Paul A. Samuelson. 1938. A note on the pure theory of consumer's behaviour. *Economica*, 5, 17, 61–71. doi:10.2307/2548836.

[17] OpenAI. 2026. API pricing: standard short-context rates. https://developers.openai.com/api/docs/pricing. Rates recorded 2026-08-28. (2026).

[18] Michael Fell, Elena Cabrio, Elmahdi Korfed, Michel Buffa, and Fabien Gandon. 2020. Love me, love me, say (and write!) that you love me: enriching the WASABI song corpus with lyrics annotations. In *Proceedings of the Twelfth Language Resources and Evaluation Conference (LREC 2020)*. European Language Resources Association, Marseille, France, 2138–2147. https://aclanthology.org/2020.lrec-1.262/.

[19] Michel Buffa et al. 2021. The WASABI dataset: cultural, lyrics and audio analysis metadata about 2 million popular commercially released songs. In *The Semantic Web (ESWC 2021)* (Lecture Notes in Computer Science). Vol. 12731. Springer, 515–531. doi:10.1007/978-3-030-77385-4_31.

[20] Spotify. 2026. About Spotify. https://newsroom.spotify.com/company-info/. Over 100 million tracks. Accessed 2026-08-31. (2026).

[21] MusicBrainz. 2026. Database statistics: languages and scripts; countries. https://musicbrainz.org/statistics/languages-scripts. Accessed 2026-08-31. (2026).