

LyricLure: Mining Catchy Hooks in Song Lyrics to Enhance Music Discovery and Recommendation

Siddharth Sharma
eshasidd@amazon.com
Amazon Inc
Sunnyvale, CA, USA

Akshay Shukla
akshukl@amazon.com
Amazon Inc
Sunnyvale, CA, USA

Ajinkya Walimbe
wajinkya@amazon.com
Amazon Inc
Sunnyvale, CA, USA

Tarun Sharma
tarunsh@amazon.com
Amazon Inc
Sunnyvale, CA, USA

Joaquin Delgado
dejoaqui@amazon.com
Amazon Inc
Sunnyvale, CA, USA

ABSTRACT

Music Search encounters a significant challenge as users increasingly rely on catchy lines from lyrics to search for both new releases and other popular songs. Integrating lyrics into existing lexical search index or using lyrics vector index pose difficulties due to lyrics text length. While lexical scoring mechanisms like BM25 are inadequate and necessitates complex query planning and index schema for long text, text embedding similarity based techniques often retrieve noisy near-similar meaning lyrics, resulting in low precision. This paper introduces a proactive approach to extract catchy phrases from song lyrics, overcoming the limitations of conventional graph-based phrase extractors and deep learning models, which are primarily designed for extractive summarization or task-specific key phrase extraction from domain-specific corpora. Additionally, we employ a multi-step mechanism to mine search query logs for potential unresolved user queries containing catchy phrases from lyrics. This involves creation of word and character *k-gram index* for lyric chunks, careful query and lyrics domain-centric normalization (and expansion) and a re-ranking layer incorporating lexical and well as semantic similarity. Together these strategies helped us create a high retrieval source specifically for serving lyrics intent queries with high recall.

KEYWORDS

key phrase extraction, span detection, phrase search, hybrid index, embedding based retrieval, natural language processing

ACM Reference Format:

Siddharth Sharma, Akshay Shukla, Ajinkya Walimbe, Tarun Sharma, and Joaquin Delgado. 2024. LyricLure: Mining Catchy Hooks in Song Lyrics to Enhance Music Discovery and Recommendation. In *18th ACM Conference on Recommender Systems (RecSys '24)*, October 14–18, 2024, Bari, Italy. ACM, New York, NY, USA, 3 pages. <https://doi.org/10.1145/3640457.3688049>

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).

RecSys '24, October 14–18, 2024, Bari, Italy

© 2024 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-0505-2/24/10

<https://doi.org/10.1145/3640457.3688049>

1 INTRODUCTION

"It's the hook that brings you back", sang *Blues Traveler* in their iconic song "*The Hook*"¹. In the realm of music discovery and search, users often turn to memorable lines or phrases from songs to find new releases or rediscover classics. However, integrating a long text field like lyrics in a heterogeneous entity like Music track that has other short text fields like track title, artist name etc. is technically very challenging in Lucene based index. Ekanayake et al. [4, 5] and [6] proposed using hybrid indexing approaches combining Lucene with relational databases to improve query performance. [21] highlights that Lucene's scoring function [15] can lead to lower ranking for longer fields due to field normalization factors, also the ambiguity and variability in BM25 implementation across systems [22] can impact retrieval performance for heterogeneous data with long text fields. These studies highlight the importance of meticulous query planning, specialized indexing and optimized scoring functions to effectively handle heterogeneous entities with large text fields.

Similarly embedding based retrieval for lyrics can face significant challenges. As highlighted in [23] and [13], longer text sequences can introduce noise, potential loss of context and semantic drift. Recent studies [7] indicate that song lyrics have become simpler and more repetitive over the last fifty years, particularly in pop and rap. This trend of using common vocabulary leads to heavy noise in retrieval if vector search is used. We even investigated the efficacy of several variants of sentence transformer[19] models² e.g. Roberta[14], XLM-R[2] etc. pre-trained on *text entailment* or *paraphrasing* objective, as well as open domain question answer models like Dense Passage Retriever(DPR) [12], using varying sizes of lyric segments(chunks). Our findings indicate significant noise in retrieval, with these models consistently failing to retrieve results containing lyrics that match the exact query phrases, necessitating alternative approaches for effective information retrieval from song lyrics.

To address the multiple challenges discussed above, this paper proposes a framework comprising of two key components, a key phrase extraction mechanism that helps us proactively mine catchy phrases from song's lyrics that we anticipate beforehand to be part of user search queries and an offline search query logs mining technique to find candidate tracks for unresolved queries that had lyrics as their intent. Our integrated approach provides a comprehensive mechanism for making songs searchable by their lyrics.

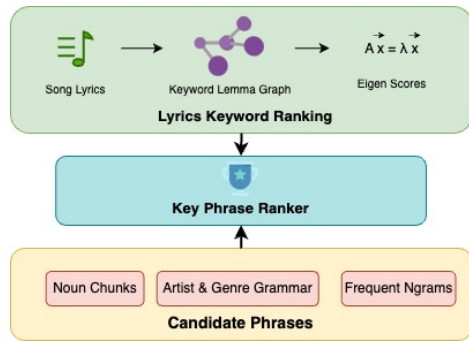


Figure 1: Lyrics Key Phrase Extractor

2 SYSTEM DESIGN

To address the previously outlined challenges we propose development of a custom solution that employs a mixture of strategies to curate a list of high-quality candidate catchy phrases.

2.1 Unsupervised Key Phrase Extraction

Catchy phrase extraction from song lyrics is challenging due to the lack of specialized datasets and the poor adaptability of domain-specific pre-trained models like *SpanBert* [10] and *keyphrase extraction models*⁴ [9]. Figure 1 shows our two pronged approach to discover catchy phrases in lyrics.

2.1.1 Candidate Key Phrase Generation. To enhance recall, we employ a multi-faceted approach for creating a list of candidate key phrases.

- (1) High Frequency Ngrams: In various musical genres, catchy phrases frequently recur in lyrics. We apply frequency-based ngram extraction and multi-layer heuristic pruning to identify these recurring patterns.
- (2) Noun Chunks: We leverage *Spacy*'s⁵ noun chunk extractor to identify and extract noun phrases.
- (3) Grammar Curation: We employ in-context learning [3] from *Large Language Models*⁶ to generate artist and genre specific grammars [1] for the extraction of phrases that better reflect their unique styles and characteristics.

2.1.2 Lyrics Keyword Graph. Off-the-shelf keyword graph libraries like *pytextrank*³ utilizing TextRank [17] and variants [8, 20] create keyword lemma graphs from words with noun, pronoun, or verb POS tags, catering to extractive summarization. Our approach employs a custom tailored keyword lemma graph creation, including words with predefined POS tags prominent in songs. We adopt TextRank's [17] methodology for word scoring, phrase scoring, and graph construction.

2.2 Search Query Logs Mining

Our approach maps high-volume, unresolved queries with potential lyrical intent to relevant tracks, ensuring accurate and user-aligned results for repeat queries. To counter the previously discussed challenges of using lexical and embedding based indices, we developed an offline mechanism that uses *hybrid index* build on lyrics chunks [11] comprising of lexical K-gram[16, Section 3.3.3] and a character n-gram index. While word k-gram index proved effective in retrieving lyrics chunks with exact n-grams as search query, character

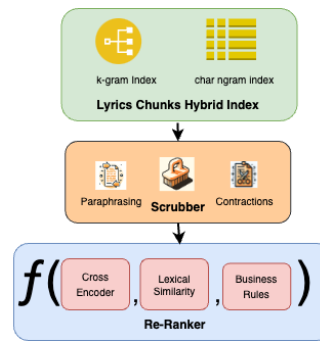


Figure 2: Query Search Log Mining

n-gram index boosted recall when query tokens were not in correct order or near similar words were used e.g. *betcha* instead of *bet you*.

Figure 2 shows the high level retrieval and ranking workflow. To improve lexical relevance we further apply a *Scrubber* layer that handles various normalization like *contractions*⁷, *paraphrasing* e.g. *"wanna"* to *"want to"* etc. Retrieved results are re-ranked by using scoring mechanism that boosted precision by combining lexical similarity, semantic similarity from a *cross encoder*⁸ model and incorporating various business critical rule.

3 RESULTS

Table 1 presents the recall@k^* metric for the methods described in section 2.1 and 2.2 relative to benchmark *Bags Of Words(BOW)*¹² performance[†]. To evaluate the recall performance of *Log Mining* method described in 2.2, we indexed [18] dataset that comprises of lyrics of tracks from 1950 to 2019.

Method	Recall@1	Recall@3	Recall@5
Key Phrase Extraction	32.14%	44.28%	64.28%
Log Mining(LM) Retrieval	66.42%	73.92%	82.85%
LM Retrieval + Re-ranking	87.85%	86.42%	88.21%

Table 1: Recall@K Metrics for Lyrics Retrieval Techniques

Due to the absence of benchmark datasets for *catchy lyrics phrases*, we curated data for *catchy phrases* in lyrics of popular tracks from various crowd-sourced *Reddit threads*^{10, 11} and enhanced its quality by manually verifying that other major music streaming platforms were able to find relevant tracks for the given queries.

4 CONCLUSION AND NEXT STEPS

Our consolidated approach, which merges unsupervised key phrase extraction with search logs mining, effectively enhances song discovery. This method is particularly effective for phrase search scenarios where there are constraints on resources such as training data and technical capacity, allowing for the rapid development of a high-performing system with limited investment. Future improvements should focus on tuning size of the lyrics chunks and refining the *Scrubber* layer to better handle the nuances of lyric search. Additionally efforts should be made to develop *LLM Agents*⁹ that can develop better artist and genre grammar and detect catchy phrase spans in lyrics.

* recall@k metric = proposed method recall minus bags of words recall

†BOW recall is evaluated using dot product similarity score of lyrics and catchy phrase one hot encoded vectors

5 BIO

Siddharth Sharma is a senior applied scientist at Amazon Music. He earned his Master's degree from North Carolina State University, Raleigh, NC. Siddharth's work focuses on search, retrieval and ranking models.

Akshay Shukla Akshay is a Software Development Engineer at Amazon Music. He earned his Master's degree from the University of Southern California in Los Angeles. Akshay works on building a platform for data scientists, aimed at providing the most relevant search results to customers.

Ajinkya Walimbe Ajinkya Walimbe is a Senior Technical Product Manager at Amazon Music. His work mainly focuses on search experience for Amazon Music customers on their mobile apps and alexa devices. Ajinkya has completed his MBA from NYU, New York and B.Tech in IT from VJTI, Mumbai, India.

Tarun Sharma is an Applied Science manager leading Search Relevance for Amazon Music. He has more than ten years of experience building and leading science and engineering teams to launch internet scale machine learning solutions and high performance systems in Ads and Recommendation domains at Amazon and previously at Microsoft. He holds a Masters in Language Technologies from Carnegie Mellon University, Pittsburgh, USA and a Bachelors in Computer Science from Indian Institute of Technology, Varanasi, India.

Joquin Delgado is currently serving as a Sr. Software Development Manager at Amazon, leading Amazon Music's Search Platform team. Previously he held Director positions at Groupon, Verizon and Intel and was the CTO of AdBrite and Co-founder/CTO of LendingClub. He also worked at Yahoo! and Oracle. His expertise lies in distributed systems, advertising technology, machine learning, recommender systems, and information retrieval. He holds a Ph.D. in Computer Science and Artificial Intelligence from Nagoya Institute of Technology, Japan and a Computer Engineering degree from Universidad Simon Bolivar, Venezuela

REFERENCES

- [1] Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly Media, Inc. <https://www.nltk.org/book/ch07.html>
- [2] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (Eds.). Association for Computational Linguistics, Online, 8440–8451. <https://doi.org/10.18653/v1/2020.acl-main.747>
- [3] Qingxiu Dong, Jingfeng Zhu, Zhenhui Zhang, Ying Xia, Ting Xiao, and Chengqing Zong. 2023. A Survey on In-context Learning. *arXiv preprint arXiv:2301.00234* (2023).
- [4] Jaliya Ekanayake, Huizhi Liang, Bing Lin, Anand Sivasubramaniam, Gautam Das, and Divyakant Agrawal. 2008. Indexing Heterogeneous XML with Pruffer Sequences. In *2008 IEEE 24th International Conference on Data Engineering*. IEEE, 126–135.
- [5] Jaliya Ekanayake, Huizhi Liang, Anand Sivasubramaniam, Gautam Das, and Divyakant Agrawal. 2010. Efficient Indexing of Heterogeneous XML Data. In *Proceedings of the VLDB Endowment*, Vol. 3. VLDB Endowment, 1072–1083.
- [6] Jaliya Ekanayake, Huizhi Liang, Anand Sivasubramaniam, Gautam Das, and Divyakant Agrawal. 2010. Indexing Heterogeneous XML Data with Relational Databases. In *2010 IEEE 26th International Conference on Data Engineering (ICDE 2010)*. IEEE, 1–12.
- [7] M. Fell and C. Sporleder. 2024. Song lyrics have become simpler and more repetitive over the last five decades. *Scientific Reports* 14 (2024), 12345. <https://doi.org/10.1038/s41598-024-55742-x>
- [8] C. Florescu and C. Caragea. 2017. PositionRank: An Unsupervised Approach to Keyphrase Extraction from Scholarly Documents. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Vancouver, Canada, 1105–1115. <https://doi.org/10.18653/v1/P17-1102>
- [9] Sujatha Gupta, Amir Pouran Ben Veyseh, Franck Dernoncourt, Thuy Nguyen, Antonis Anastasopoulos, and Jing Gao. 2021. Learning Rich Representation of Keyphrases from Text. *arXiv preprint arXiv:2112.08547* (2021).
- [10] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving Pre-training by Representing and Predicting Spans. *Transactions of the Association for Computational Linguistics* 8 (2020), 64–77. https://doi.org/10.1162/tacl_a_00300
- [11] Greg Kamradt. 2023. Semantic Chunking for LLM Applications. *arXiv preprint arXiv:2303.17770* (2023).
- [12] Vladimir Karpukhin, Sewon Min, Lyle Wu, Oyvind Tafjord, Karen Hambardzumyan, Joe West, Jason Yosinski, and Marc'Aurelio Ranzato. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2643–2653.
- [13] Omar Khattab and Matei Zaharia. 2020. Colibri: Lightweight automated machine learning for embedding-based retrieval. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. 1289–1304.
- [14] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692* (2019).
- [15] Apache Lucene. 2008. Lucene's Practical Scoring Function. https://lucene.apache.org/core/8_7_0/scoring.html. In *Apache Lucene - Scoring*.
- [16] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press. <https://nlp.stanford.edu/IR-book/html/htmledition/k-gram-indexes-for-wildcard-queries-1.html>
- [17] R. Mihalcea and P. Tarau. 2004. TextRank: Bringing Order into Texts. In *Proceedings of EMNLP-04 and the 2004 Conference on Empirical Methods in Natural Language Processing (Barcelona and Spain)*.
- [18] Luan Moura, Emanuel Fontelles, Vinicius Sampaio, and Mardônio França. 2020. Music Dataset: Lyrics and Metadata from 1950 to 2019. <https://doi.org/10.17632/3f9vbwxgr5.2>
- [19] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 3982–3992. <https://doi.org/10.18653/v1/D19-1410>
- [20] S. Rose, D. Engel, N. Cramer, and W. Cowley. 2010. *Automatic Keyword Extraction from Individual Documents*. John Wiley and Sons, Ltd, 1–20. <https://doi.org/10.1002/9780470689646.ch1>
- [21] Progress Software. 2019. Troubleshooting Lucene Search Issues. <https://www.progress.com/blogs/troubleshooting-lucene-search-issues>.
- [22] Andrew Trotman, Antti Puurula, and Elaine Ruckhaus. 2020. Which BM25 do you mean? A large-scale reproducibility study of scoring variants. *European Conference on Information Retrieval* (2020), 595–611.
- [23] Xingxing Zhang, Furu Ren, Yue Luan, Guorui Zhu, Hao Zhou, Xiaoqiang Zeng, Pengtao Xie, and Maosong Sun. 2020. Revisiting embedding based text retrieval with hierarchical segmentation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 6214–6224.

¹<https://www.youtube.com/watch?v=pdz5kCaCRFM>

²<https://huggingface.co/sentence-transformers>

³<https://pypi.org/project/pytexttrank/>

⁴<https://huggingface.co/ml6team/keyphrase-extraction-kbir-inspec>

⁵<https://spacy.io/usage/linguistic-features>

⁶https://en.wikipedia.org/wiki/Large_language_model

⁷[https://en.wikipedia.org/wiki/Contraction_\(grammar\)](https://en.wikipedia.org/wiki/Contraction_(grammar))

⁸<https://huggingface.co/cross-encoder/stsb-roberta-base>

⁹<https://www.microsoft.com/en-us/research/project/autogen/>

¹⁰https://www.reddit.com/r/TaylorSwift/comments/z5ttup/what_are_you_favorite_short_phrases_from_lyrics/

¹¹https://www.reddit.com/r/Music/comments/15vui9e/what_do_you_think_is_one_of_the_best_single_line/

¹²https://en.wikipedia.org/wiki/Bag-of-words_model