Text S3

Analysis of Song Lyrics

After obtaining the lyrics of the songs played in our study, we applied a text-mining pipeline to describe them in terms of objectively computed characteristics. We provide the full code for these analyses in our project repository under https://osf.io/x7dar/.

First, we cleaned the lyrics from formatting issues (e.g., annotations like "chorus" or missing text repetitions indicated by "2x"). Then, we combined two language detection algorithms (Joulin et al., 2016; Lui, 2016) to create a variable indicating whether a song was in English, German, or another language. We filtered for English and German lyrics (96 % of all songs), which were most likely understood by our German student sample, to ensure that our variables reflected conscious lyrics preferences. Finally, to enable natural language modeling, we translated German lyrics into English using the neural-network-based translation software DeepL (DeepL GmbH, n.d.). We describe the lyrics preprocessing in greater detail in our repository.

Next, we applied different natural language models, including Latent Dirichlet Allocation (LDA). We trained and evaluated LDA models on a separate corpus of over 180.000 lyrics from the Million Song Dataset (MSD; Bertin-Mahieux et al., 2011) to avoid overfitting the topic distributions to sample-specific patterns. After preprocessing the MSD corpus in parallel to our original lyrics file (s. details in the repository), we fit 12 topic models differing in the choice of priors and the number of topics. We tested a fixed default value of .01 for the priors alpha and beta against an optimization of the prior alpha that allows some topics to be more prominent than others. These two settings were paired with six choices for the number of topics: 15, 30, 50, 60, 75, 120. We evaluated the resulting models and chose the winner settings such that the topic coherence measure *u_mass* was maximized (Rehurek & Sojka, 2010). The winner model with fixed priors and 30 topics (s., Table S3) served to infer

topic distributions on our initial lyrics corpus. In Table S4, we provide the keywords for the final topic solution.

Another language model we applied to our lyrics corpus was a pre-trained implementation of Bidirectional Encoder Representations from Transformers (BERT) by Wolf et al. (2020). BERT provided one embedding vector for each word in a song's lyrics, plus one additional [CLS]-token embedding, which we used as a condensed numerical representation of the entire lyrics. However, 10% of our lyrics exceeded BERT's maximum input of 512 words per sequence, so we developed a cutting heuristic. When the excess words made up less than half of a song's lyrics, we followed the standard procedure to cut off the lyrics' tail which often contained fade-outs or chorus repetitions. In the remaining cases, we parsed the lyrics into multiple chunks, extracted separate embedding vectors for each chunk, and averaged them later.

The reproducible code for all lyrics analyses is available in our online repository. Single functions have been adapted from other authors and highlighted within the code for lyrics preprocessing (McKew, 2020), LDA modeling (Konrad, 2016), and song-level variable extraction (Bertin-Mahieux, 2011; The Hugging Face Team, 2020).

References

Bertin-Mahieux, T. (2011, March 21). Lyrics to bow. *Million Song Dataset, GitHub repository*. https://github.com/tbertinmahieux/MSongsDB/blob/master/Tasks_Demos /Lyrics/ lyrics_to_bow.py

Bertin-Mahieux, T., Ellis, D. P.W., Whitman, B., & Lamere, P. (2011). The Million Song Dataset. In A. Klapuri, & C. Leider (Eds.), *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011)* (pp. 1-6). University of Miami. https://doi.org/10.7916/D8NZ8J07

DeepL GmbH. (n.d.). *DeepL API* [computer software]. https://www.deepl.com/de/docs-api/

Joulin, A., Grave, E., Boianowski, P., & Mikolov, T. (2016). *Bag of tricks for efficient text*

classification. arXiv. https://doi.org/10.48550/arXiv.1607.04606

Konrad, M. (2016, June 17). Creating a sparse document term matrix for topic modeling via lda. *WZB Data Science Blog*. https://datascience.blog.wzb.eu/2016/06/17/creating-a-sparse-document-term-matrix-for-topic-modeling-via-lda/

Lui, M. (2016). langid (Version 1.1.6) [Python package]. https://pypi.org/project/langid/

McKew, J. (2020, May 29). Translating Text in Python - DeepL Translator. *Jack McKew's Blog*. https://jackmckew.dev/translating-text-in-python.html

Rehurek, R., & Sojka, R. (2010). Software framework for topic modelling with large corpora. In R. Witte, H. Cunningham, J. Patrick, E. Beisswanger, E. Buyko, U. Hahn, K. Verspoor, & A. R. Coden (Eds.), *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks* (pp. 45-50). University of Malta.

The Hugging Face Team. (2020). Transformers. *Hugging Face*. https://huggingface.co/docs/transformers/index

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., ... & Rush, A. M. (2020). HuggingFace's Transformers: State-of-the-Art Natural Language Processing. In Q. Liu, & D. Schlangen (Eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing* (pp. 38-45). Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-demos.6