find these gains in performance even when the synthetic errors are qualitatively different than the natural errors; that is, a GEC method can still perform well even when it is trained on grammatical errors that a human would never make. However, when a large dataset of natural errors is available, the improvements obtained by augmenting with synthetic errors are significantly diminished.

3. METHODOLOGY

In developing our method, we aim for it to have the following qualities:

- **High Recall Rate**: Our detection method should ideally make *no* false negatives; that is, whenever an error is present in its input, it highlights it. It is acceptable for the precision to be significantly lower than the recall as long as the user is assured that no errors lie outside the flagged regions.
- **Localization of Errors**: Within the constraint of having a high recall rate, the method should try to be as specific as possible when pointing out errors.
- **Trainable on a Small Dataset**: Many of the most common use-cases of OMR are on genres that do not have high-quality, large, digitized datasets on which we could train.
- **Takes Long Inputs**: Many errors introduced by OMR are unlikely to be identifiable in isolation from a small snippet of music, and may only be recognized on the basis of comparison with the rest of the piece.

3.1 Data Representation

What is "an error" in a musical score? Rigorously defining this notion requires some procedure to take the "difference" between two scores (in our case, an OMR output and a correct score). The characteristics of the errors that we find will depend heavily on the underlying representation we use. We also face the constraint that whatever representation we use must be suitable for input into a machine-learning model, preferably as a sequence; this disqualifies the use of methods that operate on hierarchically-structured MusicXML data [17, 18].

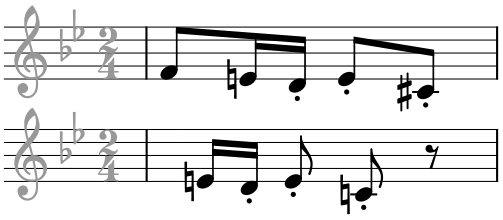
A piano roll-like representation, representing a snippet of music as a two-dimensional image, easily admits a difference operation by way of taking the pixel-wise difference of two two-dimensional piano rolls, but is memory-intensive for long inputs. Another possibility would be to use a token-based representation such as NoteTuple [19], which represents polyphonic music as an ordered list of multi-valued tokens; for example, a piece might be represented as a sequence of triples of the form `(MIDI Pitch, Onset Time, Duration)`. This kind of representation is common in deep learning applications, and there exist many notions of difference on one-dimensional sequences that could be adapted. However, to align with our goal of localizing errors, it would be better for each unit of our representation to contain as little information as possible, so that in calling a single element erroneous we pinpoint the location of the error down to a smaller region in the score.

To accomplish this, and to choose a representation that matches the domain of OMR, we opt to use an *agnostic representation* [20]. An agnostic representation of a musical score lists each of the symbols on a staff without considering their underlying musical meaning; this stands in contrast to *semantic representations* like MusicXML. An agnostic representation encodes less information per sequence element; a dotted eighth note with an accidental would typically be encoded by a single token in a representation like NoteTuple, whereas the accidental, the note, and the dot would each be a separate element in an agnostic representation, and each could be individually flagged as an error.

We write our own utility (included with the source code of this project) to convert Musicxml or Humdrum **kern files into an agnostic encoding by first parsing them with the `music21` package, and then heuristically defining an agnostic encoding that matches the resulting textttmusic21 stream. The encoding lists each symbol on a staff in order from left to right. When two or more symbols are in the same horizontal position (such as when multiple notes sound simultaneously) we list them in ascending order and add a caret symbol ^ between each of them. To handle multiple staves, we interleave them: a measure of the first violin, a measure of the second violin, the viola, the bass, and repeat. This scheme cannot encode all polyphonic musical scores into unique agnostic representations; staff lines where two rhythmically distinct polyphonic voices are notated on the same staff may contain unresolvable ambiguities, where it is not clear which notes belong to which voice. This is not an issue for our task, as we aim not to generate valid semantically-encoded scores from agnostic representations but only to *classify* notes in an existing score.

3.2 Sequence Alignment

To define a difference between two agnostically-encoded musical excerpts, we use the Needleman-Wunsch (NW) sequence alignment algorithm [21]. This algorithm takes as input two sequences and computes the shortest list of operations necessary to turn one into another, where an operation is defined as insertion, deletion, or replacement of a sequence element. By comparing a corrected score with its OMR'ed version, we can see *where* operations must be performed in order to correct it. Our strategy will be to train a model to answer the question: *Given a musical score that contains errors, where would we need to perform operations to correct those errors, according to a NW alignment?* We ignore information on *which* operations need to be performed. Preliminary experiments showed degradation in overall performance when our model was trained to identify the type of error along with its position. We surmise that to a human corrector, reliable detection of the location error itself is more important.



(a)

Original	OMR	Op.
barline.regular	barline.regular	-
8th.pos2.startBeam	–	I
accid.pos1.natural	accid.pos1.natural	-
16th.pos1.continueBeam	16th.pos1.startBeam	R
dot.pos-2	dot.pos-2	-
^	^	-
16th.pos0.endBeam	16th.pos0.endBeam	-
dot.pos-1	dot.pos-1	-
^	^	-
8th.pos1.startBeam	8th.pos1.noBeam	R
accid.pos-1.sharp	accid.pos-1.natural	R
dot.pos-3	dot.pos-3	-
^	^	-
8th.pos-1.endBeam	8th.pos-1.noBeam	R
–	8th.rest	D
barline.regular	barline.regular	-

(b)

Figure 2: (a): A correct score (top) and version with OMR errors (bottom): Mendelssohn's Quartet No. 1 in E♭ Major Op. 12, Mvt. 2, measure 10, 2nd violin part. (b): The Needleman-Wunsch alignment between the agnostic encodings of the two scores on the left. The **Op.** column shows the operations needed to correct the OMR column: **I**nsertion, **R**eplacement, and **D**eletion. Note that during training, all these operations will simply be marked as Errors.

Figure 2a contains an illustration of two snippets of music: one original and one with OMR errors. Figure 2b shows the result of their alignment with the NW algorithm. Our training data will mark each note as an error if it has an operation assigned to it; for example, the eighth rest at the end of the OMR snippet in Figure 2a will be marked as an error in training, since it must be removed to correct the score. Under this scheme we have no way of flagging a note as an error if it is not present in the OMR output. As a workaround, we flag a symbol as erroneous in training if, to correct the input, it would be necessary to insert one or more items *after* that symbol in the sequence. For example, we would mark the very first barline of the OMR output in Figure 2b as an erroneous symbol, since there is a missing `8th.pos2.start` immediately succeeding it.

3.3 Dataset and Data Augmentation

Ideally, our dataset would comprise a large, well-curated corpus of symbolic music in a single genre, each with accompanying score files in some image format, and another dataset containing versions of those score images passed through an OMR process. We could match up each symbolic music file with its OMR'ed counterpart and perform an NW alignment between them. Then, during training, the OMR'ed symbolic music would be the input to our model, and the resulting alignment would be our target.

To our knowledge, there are no large, publicly available sources of data that match these requirements. Datasets intended for training OMR systems do not include erroneous OMR outputs, as their purpose is to aid in developing the OMR process itself. Given a set of scores in symbolic and image formats, we could automate a OMR application to process a new dataset in this way, but the commercial OMR software available to us does not have the requisite batch processing functionality that would make this feasible. Instead, we take a small dataset of symbolic music files paired with their OMR'ed counterparts, and supplement it with a large dataset containing music of the same genre, but without accompanying OMR'ed files. We add synthetic errors to this larger dataset to mimic the natural errors in the small dataset.

Our main dataset, from which we derive natural OMR errors, is a set of Mendelssohn's string quartets comprising 24 movements [22], containing a total of 175 thousand tokens when encoded agnostically. This dataset contains OMR'ed versions of each movement generated using the PhotoScore software, partially-corrected versions of the OMR'ed files for each movement, and fully-corrected versions of each movement. The OMR'ed files contain a high proportion of errors; the character error rate between the fully-corrected and uncorrected files is 0.29, and the same statistic for the partially-corrected files is 0.11 [2] . We supplement this with a corpus of the string quartets of Beethoven, Haydn, Mozart, and Schubert, taken from KernScores [3] and the Annotated Beethoven Corpus [23], comprising a total of 361 movements and a total of 9.25 million agnostic tokens. It would be ideal to focus specifically on a single era of music, but we posit that for the sake of ease of training, consistency in instrumentation (restricting ourselves to only string quartets) is a more important consideration.

To augment the dataset, we add errors to the agnostic representation of our dataset in a way that mimics how natural OMR errors are distributed in the smaller dataset. To this end, we set aside a small portion of the Mendelssohn string quartet dataset and run a NW alignment between these snippets and their manually-corrected versions. From these alignments we can obtain statistics

[2] This statistic may sound high compared to the number of errors shown in Figure 1. This is explained by the verbosity of agnostic encodings; something that *looks* intuitively like a single error on the page may actually constitute several.

[3] `http://kern.ccarh.org/`

	Precision on raw OMR Output				Precision on partially corrected OMR			
	R = 0.80	R = 0.90	R = 0.95	R = 0.99	R = 0.80	R = 0.90	R = 0.95	R = 0.99
Baseline LSTUT	0.49	0.47	0.47	0.46	0.33	0.32	0.31	0.31
Architectures								
LSTM	0.49	0.47	0.47	0.46	0.11	0.11	0.10	0.10
Transformer	0.51	0.50	0.50	0.49	**0.14**	**0.12**	**0.12**	0.10
Synthetic Errors								
Random	0.47	0.46	0.46	0.45	0.10	0.10	0.10	0.10
50% fewer errors	0.53	**0.51**	0.50	0.46	0.10	0.10	0.10	0.09
Sequence Length								
64	0.47	0.47	0.46	0.45	0.10	0.10	0.10	0.09
128	0.49	0.47	0.47	0.46	0.10	0.10	0.10	0.09
512	0.50	0.49	0.49	0.48	0.11	0.10	0.10	0.10
1024	**0.56**	**0.51**	**0.51**	**0.50**	0.11	0.11	0.11	**0.11**

Table 1: A table comparing precision scores for given recall (R) scores on our error detection task. The "Baseline" model uses an LSTUT, synthetic errors made with our data augmentation method described in Section 3.3, and a sequence length of 256. Each section of the table changes a single variable from the baseline.

on how each type of symbol tends to be affected by the OMR process. For example, in the OMR process, the symbol `rest.quarter` remains the same with probability 0.527, is deleted with probability 0.142, is replaced with an eighth rest with probability 0.013, and so on. We use these statistics as a probability distribution, sampling from it for every single symbol in an agnostically-encoded piece, and then apply the resulting operation to each symbol.

This is a highly simplified model of how OMR introduces errors. It does not take into account the effects of adjacent symbols on one another, or how errors tend to group together on parts of a page that are badly scanned or damaged. However, we have a small amount of real OMR data to work from, which we plan to use for validation and testing, so we cannot spare much for the purpose of training the error generator. A more sophisticated error-generation model would likely overfit on such a small fraction of this dataset.

We train with the large dataset augmented with synthetic errors, reserving our dataset of Mendelssohn string quartets containing natural OMR errors for validation and testing. Each batch of training data in agnostic format is tokenized and is augmented with errors. Then, each training example in the batch is NW-aligned with its correct version (Section 3.2), and the alignment results are used to create a sequence of binary flags (error or no error) the same length as the training example. We train the model using the synthetic training data as input and these binary flags as the target. Validation and testing follow the same process but use existing OMR'ed data instead of adding synthetic errors.

4. EXPERIMENTS

As a baseline model, we employ the Long Short-Term Universal Transformer (LSTUT) [24] as an encoder-only model. This architectre consists of a 4-layer Universal Transformer [25] (a Transformer network where all the layers share the same weights) with 5 self-attention heads, bookended by two bidirectional Long Short-Term Memory (LSTM) layers in lieu of the positional encoding scheme used by vanilla Transformers. This architecture was designed specifically to learn from short- and long-term repetitive musical structure, and outperformed other variants tested on our tasks. We use the linear self-attention mechanism by Vyas et al. [26], implemented as part of their `fast-transformers` package for use with the Pytorch deep learning framework; this is a modification to the vanilla Transformer network with memory usage that scales linearly with input size instead of quadratically. The source code for our full workflow is available on GitHub [4] .

We test on two alternate architectures: a bidirectional LSTM and a vanilla Transformer. In the baseline LSTUT, the LSTM layers and attention heads all have an internal hidden dimension of 128. The LSTM network uses four layers, each with a hidden dimension of 512, and the vanilla Transformer network has 6 layers each containing 4 self-attention heads, all with a hidden dimension of 128. All models end in a fully-connected layer that reduces each sequence element down to a single dimension. We designed these architectures such that all would have a similar number of trainable parameters (around 8 million).

We also test on different input sequence lengths, and the type of synthetic errors used for data augmentation; the "Random" run applies uniformly random deletions, insertions, and replacements in the training data, and the "50% fewer" run uses our statistically-informed data augmentation scheme but only adds half as many errors, which we use to see if it improves performance on partially-corrected OMR. All models were trained with early stopping based on validation loss, using four NVIDIA P100-PCIE GPUs with a combined 48 GB of memory. When testing dif-

[4] `github.com/timothydereuse/transformer-omr-spellchecker`

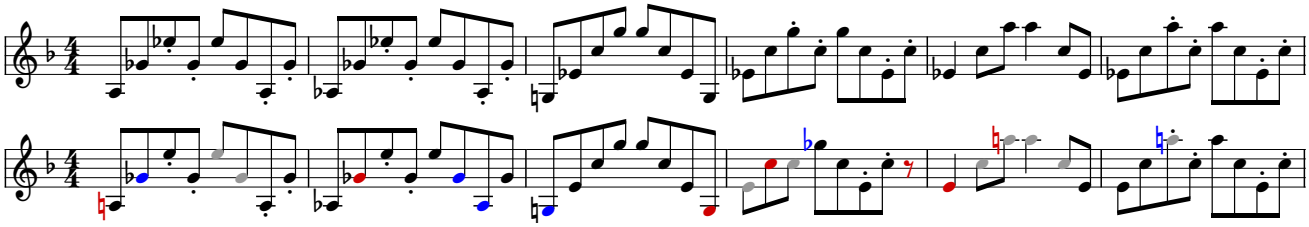


Figure 3: An excerpt from Mendelssohn's String Quartet in E♭ Major, Op. 12, Mvt 1, mm. 212-217 (1st violin part). The upper staff contains the original excerpt, and the bottom one contains OMR errors, along with the predictions of our model. Notes marked in red are errors that our model correctly identified, those marked in blue are errors that our model missed (false negatives), and those marked in grey are correct notes that our model marked as erroneous (false positives). Keep in mind that a correct note may be marked as erroneous if, to correct the score, one needs to insert a symbol *after* it.

ferent sequence lengths, we alter the batch size so as to always use the entirety of the available GPU memory. Lastly, we test on two different sets of data: one is the OMR'ed Mendelssohn quartets, and one is the same quartets after a single pass of human review. This second test set allows us to see if the model can detect the kinds of errors that humans miss when reviewing a score.

In testing, we must apply a threshold to the raw output of the network to turn it into a binary prediction. A common way to do this is to choose a threshold that maximises F1 score on the validation set and then apply that threshold to the test set. However, since we seek to maximise precision *given* a recall score near 1.0, we instead fix several scores for recall, and report the corresponding precision scores on the test set for those thresholds. We anticipate that in a real error-correction scenario, a user would be able to set a threshold themselves based on their tolerance for error; such forms of user interaction are common in high-recall information retrieval applications [27].

4.1 Results and Discussion

Table 1 shows a summary of our results. The Baseline model uses an LSTUT, synthetic errors generated with our data augmentation method, and an input sequence length of 256. We show the precision scores associated with different recall scores to represent different tolerances for missed errors. For example, our baseline model has a precision of 0.47 when the recall is 0.9, meaning that if a user wishes to be sure that 90% percent of the errors are caught, then 53% of the model's detections will be false positives.

The LSTUT model performs slightly better than the LSTM and the vanilla Transformer, though this performance difference is most notable at lower recall scores. Our statistically-informed error generation method outperforms the addition of random errors. At the highest recall score (R=0.99), our best precision score is 0.50, which we obtain at a sequence length of 1024. Precision drops only slightly as our target recall score increases, indicating that a significant fraction of each string quartet is deemed to be correct, and is ignored by the model with high confidence. Our models to perform well at identifying single errors surrounded by correct musical content, while long runs of errors, common in OMR'ed passages with tightly-spaced notes, are sometimes ignored. This may be a consequence of how our data augmentation method does not produce dense clusters of errors.

All models performed poorly on detecting errors in the partially-corrected files, though the vanilla Transformer network did marginally better. Examining our results, we note that our model struggles to notice the *absence* of elements like articulations and accidentals; these kinds of omissions comprise a majority of the errors in the partially-correct OMR files.

Figure 3 shows a violin passage from a Mendelssohn quartet and its OMR'ed version, indicating where our baseline model (at 0.90 recall) predicts errors. A common tendency of our method is illustrated here: when the model cannot pin an error to one single symbol definitively, it will mark a region around the error as erroneous (e.g., the many false positives in measures 4–5 of the excerpt). False negatives do happen (e.g., the missing staccato markings in measure 2), but when notes are missing or inserted, the model tends to make a detection that is at least in the vicinity of the true error.

5. CONCLUSION

For a machine-learning method to be useful in low-resource situations, such as those faced by many archivists and musicologists trying to make long-neglected music more accessible, it must perform a task consistently, given a small amount of data, and be easily generalizable. For this reason we have set ourselves a simple task, in hopes that we can perform it reliably. Even a low precision score may be helpful to a corrector if errors are rare; if our method flags 50% of the score as potentially erroneous, that halves the amount of material the human must check.

While our method achieves a reasonable amount of precision on Mendelssohn's string quartets, we have not proven that it can actually reduce OMR correction time. It would be useful to find ways of evaluating this type of system that speak more directly to its utility for human correctors; for example, by using the results of experiments by Pugin et al. [28] that test how long it takes to correct different types of errors. We plan to integrate our method into an existing notation application or correction interface, to see how it can best aid the symbolic encoding process.

6. REFERENCES

[1] D. Rizo, J. Calvo-Zaragoza, and J. M. Iñesta, "MuRET: A Music Recognition, Encoding, and Transcription Tool," in *Proc. of the 5th Int. Conf. on Digital Libraries for Musicology*. New York, NY: ACM, 2018, pp. 52–56.

[2] M. Alfaro-Contreras, D. Rizo, J. M. Iñesta, and J. Calvo-Zaragoza, "Omr-Assisted Transcription: A Case Study with Early Prints," in *Proc. of the 22nd Int. Society for Music Information Retrieval Conf.*, Online, 2021.

[3] A. Daigle, "Evaluation of Optical Music Recognition Software," Master's thesis, McGill University, 2020.

[4] J. Hajic Jr, J. Novotný, P. Pecina, and J. Pokorný, "Further Steps Towards a Standard Testbed for Optical Music Recognition." in *Proc. of the 17th Int. Society for Music Information Retrieval Conf.*, New York, NY, 2016.

[5] D. Byrd and J. Simonsen, "Towards a Standard Testbed for Optical Music Recognition: Definitions, Metrics, and Page Images," *Journal of New Music Research*, vol. 44, Jul. 2015.

[6] F. Rossant and I. Bloch, "Robust and Adaptive OMR System Including Fuzzy Modeling, Fusion of Musical Rules, and Possible Error Detection," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, no. 1, 2006.

[7] A. McLeod, J. Owers, and K. Yoshii, "The MIDI Degradation Toolkit: Symbolic Music Augmentation and Correction," *arXiv:2010.00059 [cs, eess]*, 2020.

[8] A. Ycart, D. Stoller, and E. Benetos, "A Comparative Study of Neural Models for Polyphonic Music Sequence Transduction," in *Proc. of the 19th Int. Society for Music Information Retrieval Conf.*, Delft, Netherlands, 2019.

[9] K. Sidorov, A. Jones, and D. Marshall, "Music Analysis as a Smallest Grammar Problem," in *Proc. of the 15th Int. Society for Music Information Retrieval Conf.*, Taipei, Taiwan, 2014.

[10] Y. Wang, Y. Wang, J. Liu, and Z. Liu, "A Comprehensive Survey of Grammar Error Correction," *arXiv:2005.06600 [cs.CL]*, p. 35, 2020.

[11] N. Madi and H. S. Al-Khalifa, "Grammatical Error Checking Systems: A Review of Approaches and Emerging Directions," in *Proc. of the Thirteenth Int. Conf. on Digital Information Management*. Berlin, Germany: IEEE, 2018, pp. 142–147.

[12] D. Naber, "A Rule-Based Style and Grammar Checker," Doctoral Thesis, Universität Bielefeld, 2003.

[13] T. Ge, F. Wei, and M. Zhou, "Reaching Human-level Performance in Automatic Grammatical Error Correction: An Empirical Study," *arXiv:1807.01270 [cs]*, 2018.

[14] C. Brockett, W. B. Dolan, and M. Gamon, "Correcting ESL Errors Using Phrasal SMT Techniques," in *Proc. of the 21st Int. Conf. on Computational Linguistics and the 44th annual meeting of the ACL*, Sydney, Australia, 2006, pp. 249–256.

[15] A. Rozovskaya and D. Roth, "Training Paradigms for Correcting Errors in Grammar and Usage," in *Human Language Technologies: The 2010 Annual Conf. of the North American Chapter of the Association for Computational Linguistics*, Los Angeles, CA, 2010.

[16] P. M. Htut and J. Tetreault, "The Unbearable Weight of Generating Artificial Errors for Grammatical Error Correction," *arXiv:1907.08889 [cs]*, 2019.

[17] F. Foscarin, F. Jacquemard, and R. Fournier-S'niehotta, "A Diff Procedure for Music Score Files," in *Proc. of the 6th Int. Conf. on Digital Libraries for Musicology*. The Hague, Netherlands: ACM, 2019, pp. 58–64.

[18] I. Knopke and D. Byrd, "Towards MusicDiff: A Foundation for Improved Optical Music Recognition Using Multiple Recognizers," in *Proc. of the 8th. Int. Society for Music Information Retrieval Conf.*, Vienna, Austria, 2007.

[19] C. Hawthorne, A. Huang, D. Ippolito, and D. Eck, "Transformer-NADE for Piano Performances," in *Proc. of the 32nd Conf. on Neural Information Processing Systems*, Montreal, Canada, 2018.

[20] D. Rizo, J. Calvo-Zaragoza, J. Iñesta, and I. Fujinaga, "About Agnostic Representation of Musical Documents for Optical Music Recognition," in *Proc. of the Music Encoding Conf.*, Tours, France, 2017.

[21] S. B. Needleman and C. D. Wunsch, "A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins," *Journal of Molecular Biology*, vol. 48, no. 3, pp. 443–453, 1970.

[22] J. Degroot-Maggetti, T. de Reuse, L. Feisthauer, S. Howes, Y. Ju, S. Kokobu, S. Margot, N. Nápoles López, and F. Upham, "Data Quality Matters: Iterative Corrections on a Corpus of Mendelssohn String Quartets and Implications for MIR Analysis," in *Proc. of the 21st Int. Society for Music Information Retrieval Conf.*, Montreal, Canada, 2020.

[23] M. Neuwirth, D. Harasim, F. C. Moss, and M. Rohrmeier, "The Annotated Beethoven Corpus (ABC): A Dataset of Harmonic Analyses of All Beethoven String Quartets," *Frontiers in Digital Humanities*, vol. 5, p. 16, 2018.

[24] J. de Berardinis, S. Barrett, A. Cangelosi, and E. Coutinho, "Modelling Long- and Short-Term Structure in Symbolic Music with Attention and Recurrence," in *Proc. of the 1st Joint Conf. on AI Music Creativity*, Stockholm, Sweden, 2020.

[25] M. Dehghani, S. Gouws, O. Vinyals, J. Uszkoreit, and Ł. Kaiser, "Universal Transformers," *arXiv:1807.03819 [cs, stat]*, 2019.

[26] A. Vyas, A. Katharopoulos, and F. Fleuret, "Fast Transformers with Clustered Attention," *arXiv:2007.04825 [cs, stat]*, 2020.

[27] H. Zhang, M. Abualsaud, N. Ghelani, M. D. Smucker, G. V. Cormack, and M. R. Grossman, "Effective User Interaction for High-Recall Retrieval: Less is More," in *Proc. of the 27th ACM Int. Conf. on Information and Knowledge Management*, Torino, Italy, 2018, pp. 187–196.

[28] L. Pugin, J. A. Burgoyne, and I. Fujinaga, "Reducing Costs for Digitising Early Music with Dynamic Adaptation," in *Research and Advanced Technology for Digital Libraries*, L. Kovács, N. Fuhr, and C. Meghini, Eds. Berlin: Springer Berlin Heidelberg, 2007, vol. 4675, pp. 471–474.

"MORE THAN WORDS": LINKING MUSIC PREFERENCES AND MORAL VALUES THROUGH LYRICS

Vjosa Preniqi[1] **Kyriaki Kalimeri**[2] **Charalampos Saitis**[1]
[1] Centre for Digital Music, Queen Mary University of London, London UK
[2] ISI Foundation, Turin, Italy
{v.preniqi, c.saitis}@qmul.ac.uk, kyriaki.kalimeri@isi.it,

ABSTRACT

This study explores the association between music preferences and moral values by applying text analysis techniques to lyrics. Harvesting data from a Facebook-hosted application, we align psychometric scores of 1,386 users to lyrics from the top 5 songs of their preferred music artists as emerged from Facebook Page Likes. We extract a set of lyrical features related to each song's overarching narrative, moral valence, sentiment, and emotion. A machine learning framework was designed to exploit regression approaches and evaluate the predictive power of lyrical features for inferring moral values. Results suggest that lyrics from top songs of artists people like inform their morality. Virtues of hierarchy and tradition achieve higher prediction scores ($.20 \leq r \leq .30$) than values of empathy and equality ($.08 \leq r \leq .11$), while basic demographic variables only account for a small part in the models' explainability. This shows the importance of music listening behaviours, as assessed via lyrical preferences, alone in capturing moral values. We discuss the technological and musicological implications and possible future improvements.

1. INTRODUCTION

The field of music recommender systems has a lot to gain from the fields of music psychology and sociology [1, 2], where researchers have found converging evidence that people listen to music that reflects their personality needs [3–7] and helps express their values [8–10]. For example, extroverted people tend to choose more energetic and rhythmic tunes, while listeners holding values of understanding and tolerance prefer more sophisticated and complex music. Indeed, operationalising knowledge of how personality traits relate to listener taste and preferences has already been shown to improve music recommendations [11, 12] and to make them more diverse [13]. Yet personality dispositions alone may not suffice to explain, and thus model, our music listening behaviours.

Aiming to advance an integrative view of the music listener, which may benefit music recommender system scenarios, we set to explore the less attended relation between moral values and music preferences. If personal values are conceived as intrinsic motivational goals, moral values reflect traits learned under the influence of society, culture, and religion, amongst others, which bond people together into groups. Considering music as an evolved tool of social affiliation and bonding [14, 15], it is reasonable to speculate that people may like certain music styles and genres because they provide stimuli that match their morality-related needs.

We further hypothesise that moral values are expressed more clearly in a verbal rather than non-verbal manner and examine their influence on musical taste through lyrics. When people listen to sung music, their preferences are driven by not only the audio content but also the content of lyrics [16]. Lyrics convey rich, multifaceted messages about societal issues such as love, life and death, but also political or religious concepts, often independently from melodic and other audio information [17]. Lyrical messages can support listeners' mental health [18]. Nonetheless, little is known about whether lyrical information manifests links between psychological traits and music preferences [19]. To what extent are moral values reflected in the lyrics of one's favourite songs? Do lyrics predict the moral traits of listeners?

To tackle these questions, we used data from the *LikeYouth.org* project, a Facebook-hosted application developed specifically for research purposes as a surveying tool and was mainly deployed in Italy. Upon providing their informed consent, participants completed validated psychometric questionnaires for personality, moral traits and basic human values, basic demographic information such as age and gender, while agreed to share their Page Likes (see [20] for a detailed description of the complete dataset). For the purpose of this study we only analysed moral values scores and Likes on music artist Pages. Combining these with information from the *genius.com* music database, we obtained the lyrics from the five most popular songs per artist. We performed both sentiment [21] and emotion [22] analysis on the obtained lyrics, assessed their moral narratives employing the MoralStrength lexicon [23], and examined themes and overarching narratives through topic modelling [24].

We built a series of regression models that infer moral traits from lyrical content, demographics, and Likes-based

features (e.g., artist popularity). Our findings show that peoples' worldviews and moral values are indeed reflected in their music preferences as modelled through lyrics, in line with recent literature [25, 26]. Extracting topic and moral features from lyrics specifically increased model performance over sentiment, emotion, and demographic features.

We contribute to the growing literature studying the interplay between music and psychology, with findings that clearly link the preferences of people to artists and songs that are in line with their moral values. Personalised recommendations for streaming on-demand music can be greatly enriched by including notions of moral worldviews about their listeners instead of only shallow psychological attributes [1, 5, 10]. Such knowledge can be directly implemented in psychologically aware music recommender systems, improving music streaming services and contributing to listener wellbeing [27]. On a different key, the relationship between moral worldviews and music preference is crucial to inform communication experts about their choice of the most appropriate music piece to accompany a social campaign.

2. BACKGROUND

We operationalise morality via the Moral Foundations Theory (MFT) [28], which expresses the psychological basis of moral reasoning in terms of five innate foundations, namely Care/Harm, Fairness/Cheating, Loyalty/Betrayal, Authority/Subversion, and Purity/Degradation. These can further collapse into two superior foundations: *Individualising* (Care and Fairness), indicative of a more liberal perspective, and *Binding* (Purity, Authority and Loyalty), indicative of a more conservative outlook.

Moral foundations are considered to be higher psychological constructs than the more commonly investigated personality traits [29]. They have been associated with attitudes towards complex situations such as politics [30, 31], climate change [32], and vaccination [33, 34].

However, moral values have attracted less attention from music scientists. Using data from an ad-hoc online survey comprising, among other items, MFQ scores and preferences ratings on 13 music genres, Preniqi et al. [25] found that people with higher levels of Binding foundations (e.g., more authoritarian individuals) tend to listen to country and Christian music, the lyrics of which often foster notions of tradition [3]. Those with lower levels of Binding traits tend to prefer music genres such as punk and hip-hop, where lyrics are known to challenge traditional values, and the status quo [8]. Individualising foundations were overall harder to predict (cf. [35]). Furthermore, including demographic information (e.g., age, gender, political views, education) improved MFT predictions marginally, indicating the ability of music preferences alone to explain one's moral values.

In the computational social science field, recent work has demonstrated the predictability of MFT traits from a variety of digital data, including gameplay [36], smartphone usage and web browsing [35]. Moral values can also be explained by verbal data, as they can be more clearly communicated through thoughts and opinions [23, 34, 37]. Several dictionary-based approaches for predicting moral values expressed in texts such as tweets and other social media posts have been proposed, including the Moral Foundations Dictionary [37, 38] and the MoralStrength lexicon [23]. Here we employ the latter to uncover moral narratives in song lyrics, which we then use to predict the moral traits of listeners.

		Census	**MFT All data** $n = 3,920$	**MFT & ≥10 Page Likes** $n = 1,386$
Gender	M	48%	54%	53%
	F	52%	46%	47%
Age	<25	23%	21%	29%
	≥25	77%	79%	71%

Table 1. Demographic breakdown of our data according to gender and age. The "Census" column reports the national distribution per attribute according to the statistics provided by the official census bureau [45].

The relation between lyrics and music preferences has only recently started to receive attention across music and social psychology disciplines. Some studies have suggested associations between the personality or mental health of songwriters and their lyrics [39, 40]. On the listener side, neurotic individuals tend to listen to songs with more complex and less repetitive lyrics that express negative emotions [19, 41]. More conscientious individuals tend to prefer lyrics talking about achievements [19] but also about love [42]. Importantly, preferences for lyrics are found to be predictive of personality traits distinctly from audio or melodic preferences [19, 42].

Concerning moral values, in recent work, they have been found to explain a unique and significant portion of the variance in the lyrical preferences of different metal music sub-genre fans that was not already accounted for by personality traits [26]. For example, preferring lyrics about celebrating metal culture and unity was related to higher levels of the Loyalty foundation and higher levels of extroversion. In U.S. popular music, an increase in lyrics related to self-focus and -promotion since the 1980s has been shown to manifest the increasing individualism of American society [43, 44].

3. DATA COLLECTION

The LikeYouth Facebook-hosted application was initially launched in March 2016, while the data used here were downloaded in September 2019. It was deployed mainly in Italy, where approximately 64,000 people entered the platform, from whom 3,920 users (90% geolocated in Italian territory) filled out the MFT questionnaire correctly.

Of those, 47% did not provide their age due to the facultative nature of LikeYouth. Because we wished to include age as a demographic predictor variable, we inferred the missing values from all (e.g., not just music artist related) Page Likes of the 3,920 users. Similar to [46] we created