A CORPUS-BASED STUDY OF RHYTHM PATTERNS

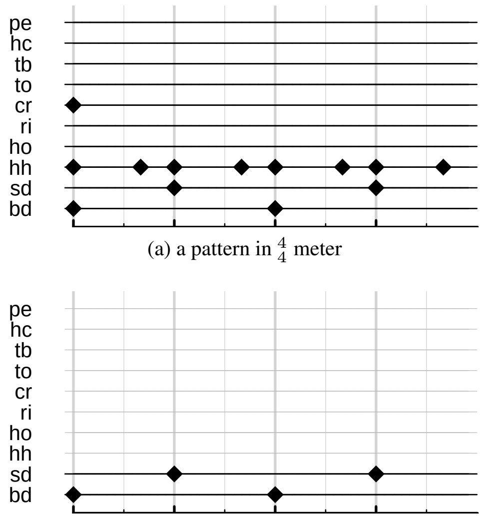
Matthias Mauch **Simon Dixon**
Centre for Digital Music, Queen Mary University of London
{matthias.mauch, simon.dixon}@eecs.qmul.ac.uk

ABSTRACT

We present a corpus-based study of musical rhythm, based on a collection of 4.8 million bar-length drum patterns extracted from 48,176 pieces of symbolic music. Approaches to the analysis of rhythm in music information retrieval to date have focussed on low-level features for retrieval or on the detection of tempo, beats and drums in audio recordings. Musicological approaches are usually concerned with the description or implementation of man-made music theories. In this paper, we present a quantitative bottom-up approach to the study of rhythm that relies upon well-understood statistical methods from natural language processing. We adapt these methods to our corpus of music, based on the realisation that—unlike words—bar-length drum patterns can be systematically decomposed into sub-patterns both in time and by instrument. We show that, in some respects, our rhythm corpus behaves like natural language corpora, particularly in the sparsity of vocabulary. The same methods that detect word collocations allow us to quantify and rank idiomatic combinations of drum patterns. In other respects, our corpus has properties absent from language corpora, in particular, the high amount of repetition and strong mutual information rates between drum instruments. Our findings may be of direct interest to musicians and musicologists, and can inform the design of ground truth corpora and computational models of musical rhythm.

1. INTRODUCTION

In Western popular music and jazz, the main percussive instrument is the drum kit, consisting of a collection of drums and cymbals arranged around the drummer. Drum kits can contain a large range of different instruments. The *bass drum* (or: *kick drum*) is usually the drum with the lowest frequency and is operated via a foot pedal. The *snare drum*, the dominant back-beat instrument, has a higher-pitched sound with additional noise components from the *snares* spanned across its lower skin. The *hi-hat* is made from two cymbals facing each other, which the drummer can open and close via a foot-pedal. The closed hi-hat has a short, high-pitched sound, whereas the open hi-hat has a longer sustain. *Ride cymbals* have a sustained high-pitched ring, while the *crash cymbals'* sound is usually more noise-like. *Tom-toms* are drums of intermediate sizes between the bass and snare drums. *Hand-claps* and *tambourine* are often used to provide additional colour, along with further varied percussion instruments, which we do not discuss in this study. The history and makeup of the modern drum kit is covered comprehensively elsewhere, e.g. [3].



(a) a pattern in $\frac{4}{4}$ meter

(b) pattern projected to the bass drum/snare drum space

Figure 1: Example drum patterns in drum tab notation, with musical time on the x axis; strong lines are beats. Drums from bottom to top: bd – bass drums, sd – snare drums, hh – hi-hat closed/pedal, ho – hi-hat open, ri – ride cymbals, cr – crash cymbals, to – toms, tb – tambourine, hc – hand clap, pe – other percussion.

In music information retrieval (MIR), most research on rhythmic features has concentrated on beats, meter and tempo. The tracking of beats establishes a temporal grid on a piece of music, which is useful to anchor other descriptors of a musical piece in time. The timing of beats also determines the tempo, which correlates with the perceived speed of the music [10]. The automatic identification of meter [9] is also based on beats, and provides additional rhythmic information. However, neither meter nor tempo, nor their combination, capture the temporal sequence of rhythmic events. Several audio features have been developed to include this temporal information [5, 17, 19] with considerable success, especially for clear-cut cases such as the classification of ballroom dances [4]. Being concerned with good performance in retrieval tasks, these methods are deliberately agnostic to how the rhythmic signal was originally created. On the other end of the spectrum lies the automatic transcription of music, and that of drums in particular. Drum transcription algorithms [7, 22, 23] usually neglect musical higher-level context such as meter and simultaneous rhythm patterns. One possible exception is Paulus's n-gram model of drum sequences [18] that informs transcription. The n-gram model demonstrates that context models can be useful to suppress errors, but it is also quite obvious that modelling rhythm as sequences of half-beat length symbols is a strong simplification that cannot capture interactions of concurrent rhythms played on multiple drums. A further simplification present in most drum transcription papers is the very small set of different drums considered: bass drum, snare drum and hi-hat. In this paper, we consider a much larger rhythm space, both in terms of temporal context and drum instrumentation.

Comparatively little work in MIR has quantitatively examined rhythms in symbolic data. While Muramaki's work on drum fill detection [15] is concerned with analysis, most work is focussed on improving music production, for example by combination drum loops of suitable complexity [21]. The study of rhythm has a long tradition in musicology, but only in recent decades has empirical music analysis found its way into the musicological tradition. Notable tools include the Humdrum Toolkit [8], jSymbolic [14] and music21 [2], which facilitate the processing of symbolic music, but do not directly examine the statistical properties of the corpus itself, nor provide tools as sophisticated as those available for natural language processing. In the domain of harmony, some attempts have been made to analyse chord progressions with language models [13, 20].

In this paper, language models are employed to analyse the statistical properties of a large corpus of drum parts, to reveal the degree of variety within and between pieces, and to discover interdependencies between different parts of the drum kit. In the next section we describe our representation of rhythm patterns, while in section 3 an overview of the data set, consisting of 48,176 MIDI files, is given. Section 4 provides the results of our analyses, and the final two sections contain a brief discussion and conclusions.

2. DRUM PATTERN DEFINITION

In order to build a corpus of drum patterns, we need to segment the music into short chunks whose lengths corresponds to meaningful metrical units. Since we are dealing with a symbolic representation which provides unambiguous onset times, the main effort required is to parse the events according to the metrical structure, suppressing performance-related information such as fluctuations in tempo, timing, and dynamics, which—for the purposes of this study—we are not interested in. Instead, similar to linguists building text corpora from *stemmed* words with grammatical endings removed, we build reduced drum pattern models by applying five levels of abstraction.

Bar segmentation. The tracks are segmented into bars as encoded in the MIDI files. Each bar is a *token*, the fundamental unit, similar to word tokens in language.

Drum categorisation. We summarise the General MIDI standard drums into 10 known drum categories (see Figure 1) and one *unknown* category.

	count	portion
4/4	4,305,516	90.3%
3/4	188,297	3.9%
2/4	114,068	2.3%
6/8	53,681	1.1%
12/8	19,575	0.4%
other	84,830	1.7%

Table 1: Distribution of time signatures in the corpus.

Tempo abstraction. We discard tempo information (but not metrical structure).

Intensity abstraction. We discard sound intensity information, i.e. MIDI velocity.

Quantisation. We quantise the drum notes relative to the beat, reducing the granularity to a grid of 12 equally spaced divisions per beat span, and retain only their onset time.

The resulting representation contains approximately the same information that would be found in traditional score notation. After this "stemming" procedure, we characterise a drum pattern via the presence (or absence) of drum onsets for each beat, position within the beat, and drum category, as visualised in the example *drum tab* representation shown in Figure 1a. Hence, a bar with N_b beats can be represented as a binary sequence of $N_b \times 12 \times 11$ bits. For the most frequent time signature, $\frac{4}{4}$, the number of beats is $N_b = 4$, and so the space of possible $\frac{4}{4}$ patterns allows $2^{4\times12\times11} \approx 10^{159}$ different patterns. Thus, despite five abstraction steps, we have retained an extremely large pattern space. Since the space is much larger than any data set, it is clear that large parts of the space will never appear in actual music. We show later that we can not only quantify the size of the space used in a given corpus, but also make predictions about how much of the space will be used as the corpus grows.

We define drum pattern sub-spaces by discarding some drums or metric positions. For example, if we restrict our attention to sub-patterns made of only bass and snare drums, a large number of different full patterns with, say, different use of the hi-hat would be mapped to the sub-pattern shown in Figure 1b.

3. DATA

We collected 72,283 unique MIDI files from the Internet. In order to understand the nature of the resulting collection, we drew a random sample of 100 songs and manually classified them. The sample mainly contains pop/rock music (62 songs), film music (10), jazz (9), classical (7) and country/folk music (6). Of the six remaining songs, five are of various genres and one was not decodable. A large proportion of the songs are good-quality renditions of popular recordings.

A study of the within-track interonset intervals (IOIs) on the whole dataset reveals that many songs are already quantised; about a third of the songs (34%) contain $> 99\%$

(a) All metrical positions

drums	# types	predicted # types at 20M tokens	$\mathcal{P}$ in %	R_{sw} in %	R_{loc} in %
all	**656798**	**1688906**	**6.23**	**73.5**	**33.4**
bd	46243	101230	0.45	91.1	64.2
sd	62647	143525	0.62	90.5	67.5
bd/sd	186688	454218	1.90	85.5	52.6
hh/ho	76351	174590	0.79	91.3	69.4
cmb	170344	415935	1.76	85.8	54.3
to	29394	70500	0.30	95.3	85.9
hc/tb/pe	84417	191712	0.88	94.4	81.9
bd/sd/cmb	466962	1176552	4.76	77.6	38.5

(b) Beats 1 and 2 only

drums	# types	predicted # types at 20M tokens	$\mathcal{P}$ in %	R_{sw} in %	R_{loc} in %
all	342453	786850	3.30	82.1	48.0
bd	7602	14788	0.07	94.9	76.8
sd	14272	30612	0.14	94.7	79.5
bd/sd	48493	106701	0.47	91.3	68.5
hh/ho	21460	44131	0.20	94.5	79.1
cmb	57287	124465	0.54	90.8	64.8
to	7523	16782	0.07	97.2	92.1
hc/tb/pe	32014	67845	0.31	96.1	86.9
bd/sd/cmb	198699	454309	1.94	85.1	52.9

Table 2: Sub-pattern statistics. $\mathcal{P}$ is productivity (see Section 4.2), R are repetition indices (Section 4.3).

IOI-quantised events, while 60% still contain $> 75\%$ IOI-quantised events. Our impression that the songs are usually carefully crafted for authentic playback is reflected in the fact that 71% of the songs have varied velocities (less than half of the notes uses the most popular velocity), i.e. it is likely that only few songs are MIDI exports from music typesetting programs.

In order to limit the influence of abnormally long songs only notes less than 20 minutes into any song are considered. Very soft drum notes (velocity < 20) are removed. We exclude songs with empty drum tracks, and those whose musical beat is likely to be out of sync with the MIDI beat (i.e. where the frequency of on-beat drum notes is $< 50\%$ that of the most frequent quantisation).

After decoding, the collection contains 4,765,947 bar tokens in 48,176 files, which corresponds to a mean of around 99 bars per song. The overwhelming majority, 90% of bars, is in $\frac{4}{4}$ time, with only a few other time signatures exceeding 1% of the corpus (see Table 1).

The terms *type* and *token* are borrowed from natural language processing and will be used here as follows:

type: unique drum pattern ($\approx$ unique word in language),

token: drum pattern type instance.

The overall number of bar types in our database is 656,798. The sub-pattern spaces retain the same number of tokens, but can have dramatically fewer distinct pattern types, as Table 2 (column 2) shows.

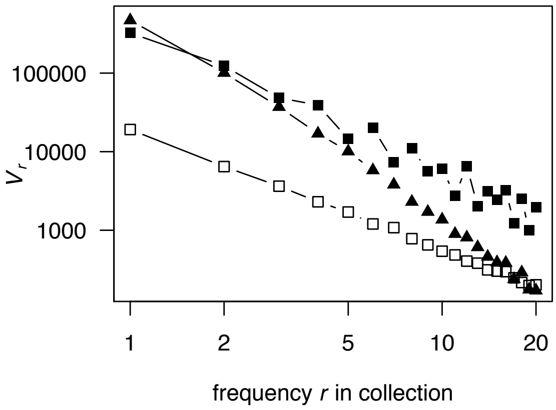


Figure 2: Type frequencies V_r by token count r for drum patterns (filled squares), drum pattern by song (filled triangles), the Brown language corpus (blank squares).

4. RESULTS

This section provides an overview and discussion of some insights that can be gained from our corpus of drum patterns. More exhaustive information is available at *http://isophonics.net/ndrum*.

4.1 Large Number of Rare Events

As with many natural language corpora, the distribution of type counts and the frequencies of these type counts are extremely skewed. Figure 2 shows a plot called *frequency-of-frequencies* plot, in which the number of type occurrences r in the database is plotted against the number of types V_r that occur r times. The figure shows three graphs: drum pattern counts, song-wise drum pattern counts (one per song in which the type occurs), and word type counts from a corpus of American English, the *Brown Corpus*. The number V_1 of types that occur only once in the whole collection is greatest; tokens occurring twice already account for much smaller fractions of the corpora, a phenomenon often referred to as *large number of rare events* (LNRE).

The log-log scale plot in Figure 2 illustrates an additional property of the data: all distributions can be approximated by a straight line, a characteristic of "scale-free" distributions. For a discussion of this phenomenon, see, for example, [16]. While the full drum pattern count (filled squares) resembles the word distribution in the Brown corpus in slope (the absolute height reflects that the Brown corpus has only 1M tokens), it is not as smooth as the Brown corpus's. However, the unstable nature of the graph is not random; rather, the higher values at multiples of 2 reflect the usual organisation of music in units of even multiples of bars. As we would expect, then, counting the number of *songs* in which a type appears leads to a much smoother graph (filled triangles) that is unaffected by repetitions. We will return to song-wise counts in Section 4.3.

4.2 Vocabulary Growth

For LNRE distributions we can estimate how fast the number of types (in our case: distinct drum patterns) is growing with vocabulary size. A popular measure for that is productivity $\mathcal{P}$ [1]. For a corpus of size N with V_1 types that occur only once,

$$\mathcal{P} = V_1/N. \tag{1}$$

This measure is an indicator of the potential to generate new patterns. The productivity of large pattern spaces is generally much higher than that of smaller sub-spaces. For example, all productivity values in Table 2b, where the sub-patterns are constrained to the first two beats of a bar, are far smaller than the respective ones in table 2a. More interestingly, however, there are also large differences to be found between single drums. For example, the productivity of the snare drum as shown in table 2a is far greater than that of the bass drum in the same table, suggesting that snare drum patterns are used more creatively (most probably due to the bass drum usually being operated by one foot). In fact, assuming a Zipf-Mandelbrot model [6], we can predict the vocabulary size as a function of corpus size; Table 2 displays productivity values and the predicted number of tokens for a vocabulary size of 20 million.

4.3 Repetition and Different Ranking Types

Simply using the relative frequencies p_{rf} of pattern types is the standard way to measure word probabilities, but it is less informative in music because of the high amount of repetition present. In our paper on chord progressions [12] we suggest to use the proportion of songs a (chord) pattern occurs in, which we call p_{sw} here. For example, counting a token only once per song reduces the overall token count from $N = 4,765,967$ to $N_{\text{sw}} = 1,264,139$ for the full pattern spaces. A softer way of reducing the influence of repetition is motivated by the observation that drum patterns in consecutive bars are often identical: one can eliminate tokens that are exact repetitions of the immediately preceding token. This locally non-repeating set has $N_{\text{loc}} = 3,176,153$ tokens, with relative frequencies denoted by p_{loc}. We use the reduced token counts to define local and song-wise repetition indices:

$$R_{\text{loc}} = 1 - \frac{N_{\text{loc}}}{N} \text{ and } R_{\text{sw}} = 1 - \frac{N_{\text{sw}}}{N}. \tag{2}$$

Table 2 lists the repetition indices (in %) for different sub-pattern corpora. Even the full patterns have a song-wise repetition index R_{sw} of 74%, meaning that only just more than a quarter of the drum patterns per song are unique. A similar picture emerges when looking at local repetition, which accounts for $R_{\text{loc}} \approx 33\%$ of all tokens. Repetition is even more dominant in smaller sub-patterns: $R_{\text{sw}} = 90\%$ of snare drum pattern tokens are repeated within a song, and $R_{\text{sw}} = 68\%$ are repetitions of the preceding bar.

In Figure 3, we show the 10 most common bar-length patterns in the corpus. Empty patterns with different time signatures occupy the 1st, 4th and 5th rank, while standard rock patterns using only bass, snare and closed high-hat occupy the remaining ranks. A variation with a swing high-hat pattern appears at rank 9.

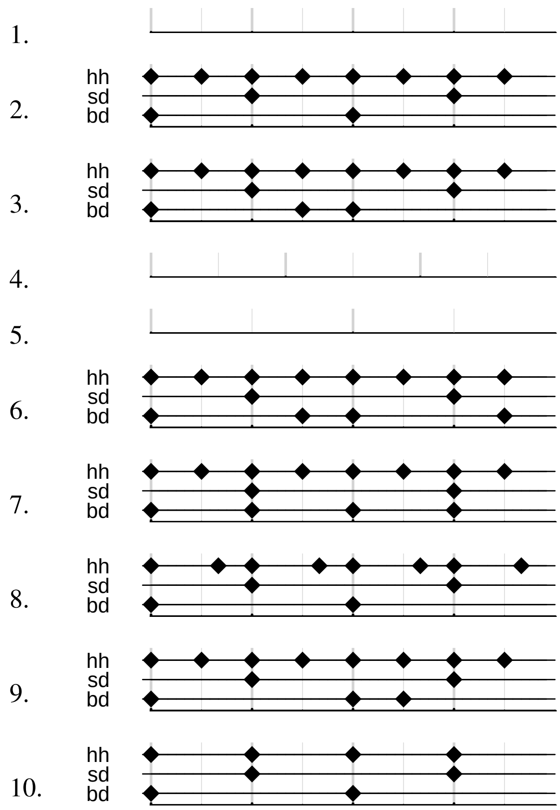


Figure 3: Ranking by p_{rf} score (Section 4.3).

The song-wise results are shown in Figure 4, with the empty patterns removed. Although some of the same patterns appear, the non-empty pattern which occurs in the most songs is a crash cymbal and bass drum on the first beat of the bar, presumably at the end of a piece or section. Rank 3 and 7 have quarter note patterns often used for "counting in" a song, while ranks 4 and 8 are the two sub-patterns of the rank 2 pattern — a single crash cymbal and a single bass drum respectively. Ranks 5 and 6 contain standard rock drum beats seen previously. The results with local repetition removed (i.e. ranked by p_{loc}) are similar and are not shown here. Comprehensive rankings can be found at *http://isophonics.net/ndrum*.

4.4 Collocations and Typical Drum Patterns

Linguists have long realised that interesting, idiomatic word combinations do not usually appear in the top ranks when sorted by frequency. *Collocations*—combinations of two words that occur more often than would be expected from their individual frequencies—are usually more interesting and meaningful. One strategy to discover collocations is to consider two hypothetical models: H_1, by which the likelihood of one of the tokens to occur depends on the other, and H_2, by which their occurrences are independent. One can then calculate the likelihood ratio

$$\log \lambda = \log \frac{L(H_1)}{L(H_2)} \tag{3}$$

of the two hypotheses for any pair of word types—or indeed drum pattern types. We follow Manning and

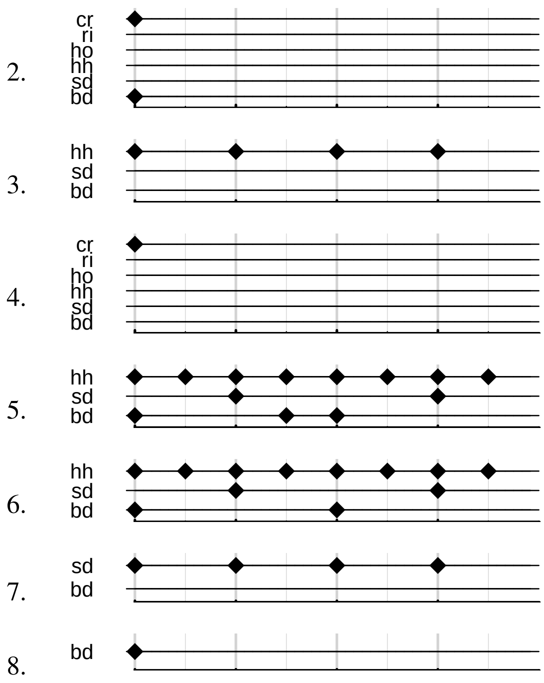


Figure 4: Ranking by p_{sw} score (Section 4.3).

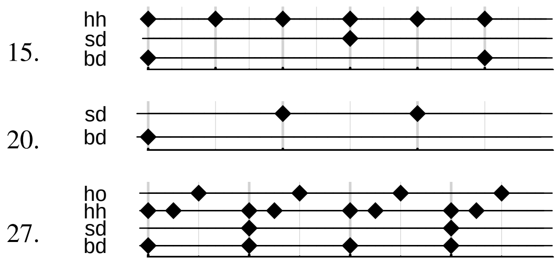


Figure 5: Ranking by collocation score (Section 4.4).

Schütze's approach [11, Chapter 5], assuming binomial type count distributions, and calculate $\log \lambda$ scores for combinations of bass drum/snare drum patterns on the one hand and hi-hat (open and closed) patterns on the other.

Ranking by the collocation score (3) results in a list of *typical* drum patterns that need not necessarily be frequent. Figure 5 shows some example of rarer patterns that nevertheless rank much higher than in the frequency rankings discussed in Section 4.3. For example, the typical $\frac{6}{8}$ pattern at rank 15 appears only at rank 99 in the raw frequency ranking (and at ranks 59 and 389 when ranked by r_{loc} and r_{sw}, respectively). The $\frac{3}{4}$ pattern at rank 20, too, is much further down the frequency rankings (48, 115, 264), as is the disco-style pattern at collocation rank 27 (35, 67, 171).

4.5 Mutual Information

That the decomposition of drum patterns is meaningful can be illustrated by the fact that the information flow between the sub-patterns across the corpus models musical relationships between them. The entropy (in bits) of a discrete

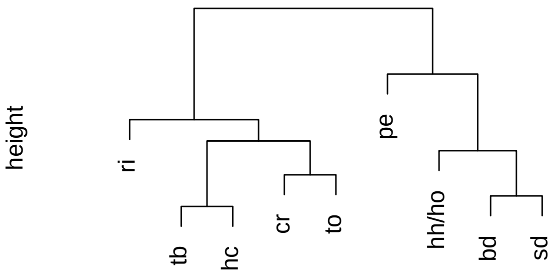


Figure 6: Hierarchical clustering of nine drum types by mutual information. The distance matrix is based on the inverted, normalised mutual information values (see text).

distribution X in with probabilities p_i is defined as

$$H(X) = -\sum_{i=1}^{N} p_i \log_2 p_i \tag{4}$$

with the convention that if $p_i = 0$, then $p_i \log_2 p_i = 0$. It expresses how much information is needed in order to represent the distribution. While this is interesting in itself, we are interested in how much information two drum pattern sub-spaces X and Y share. This is what mutual information expresses. It is defined as

$$I(X;Y) = H(X) + H(Y) - H(X,Y). \tag{5}$$

To normalise the measure we divide by the sum of the individual entropies, and to turn the similarity into a measure of divergence we take the exponential of its negation:

$$d(X,Y) = \exp\left\{-\frac{I(X;Y)}{H(X)+H(Y)}\right\}. \tag{6}$$

This allows us to calculate pair-wise divergence values between all drum types. The result is visualised in Figure 6 as a binary tree obtained by hierarchical clustering with the complete-linkage algorithm. The more information is shared between drums according to d, the closer they will appear on the tree. The algorithm has indeed recovered aspects of the usage of the drum kit, with the drums that form the core of most rhythms in popular music—bass drum, snare drum and hi-hat—grouped together on the right, loosely associated with the percussion instruments. Within the remaining drums on the left hand side, the ride cymbals have their own branch, whereas the tambourine is grouped with hand claps, and crash cymbals with tomtoms, with grouping suggesting high mutual information.

5. DISCUSSION AND FUTURE WORK

We must be aware that findings made in the MIDI domain may only partially be applicable to other music. Furthermore, the size of the database prohibits the manual verification of every song. In addition to the measures described in Section 3, automatic sanity checks could further reduce the noise in the data.

We expect that the outcomes of the present study will be valuable to musicians and researchers, so we are interested in a rigorous evaluation of its usefulness. Several scenarios are conceivable. For example, our system can easily be extended to return a ranked list of pattern synonyms, i.e. patterns that are used in similar contexts as the query pattern—a creative tool for drummers. A useful music informatics application could be to extend the promising n-gram technique for audio drum transcription proposed by Paulus, especially with models that "back off" [11, Chapter 6] not only in time, but also in the sub-pattern instrument spaces we have introduced.

6. CONCLUSIONS

We have introduced a novel method of empirical research on musical rhythm by considering bar-length drum patterns and treating them analogously to words in natural language processing. This paper has shown that the approach yields useful and interesting results because the palette of tools available from language processing can—to a large extent—be used in the musical domain, too.

We have found that the distributions of drum patterns resemble those of and English words, and have used this fact to predict vocabulary growth in our musical corpus. Vocabulary growth predictions can be useful to inform decisions on how much ground truth is needed to cover a given proportion of unseen data.

We have discovered some properties that clearly distinguish our data from language corpora, most prominently the extremely high degree of repetition. A second, more subtle, difference is that drum patterns can be decomposed in time and by instrument, yielding distributions with different characteristics.

We have proposed three simple ways of ranking drum patterns by raw frequency, repetition-reduced frequency, and song frequency. In order to identify not only frequent, but *interesting* drum pattern combinations, we have applied collocation ranking to our drum corpus. For musicians, the pattern rankings, which can be found at *http://isophonics.net/ndrum*, may be the most interesting aspect of this paper.

Finally, by calculating the mutual information flow between sub-patterns pertaining to the individual drum categories, the drum categories that are musically related cluster together.

We believe that the corpus-based study of rhythm as proposed in this paper is interesting not only to musicians. Musicologists and music informatics researchers might find it a valuable resource to obtain a quantitative view on rhythm and drum patterns.

7. REFERENCES

[1] R. H. Baayen. Quantitative aspects of morphological productivity. In Geert Booij and Jaap Van Marle, editors, *Yearbook of Morphology 1991*, pages 109–149. Kluwer Academic Publishers, 1992.

[2] M. S. Cuthbert. music21: A Toolkit for Computer-Aided Musicology and Symbolic Music Data. *Proc. of the 11th Int. Conf. on Music Information Retrieval (ISMIR 2010)*, pages 637–642, 2010.

[3] M. Dean. *The Drum: A History*. Scarecrow Press, 2011.

[4] S. Dixon, F. Gouyon, and G. Widmer. Towards characterisation of music via rhythmic patterns. *Proc. of the 5th Int. Conf. on Music Information Retrieval (ISMIR 2004)*, pages 509–516, 2004.

[5] D. P. W. Ellis and J. Arroyo. Eigenrhythms: Drum pattern basis sets for classification and generation. *Proc. of the 5th Int. Conf. on Music Information Retrieval (ISMIR 2004)*, pages 554–559, 2004.

[6] S. Evert and M. Baroni. Testing the extrapolation quality of word frequency models. *Proc. of Corpus Linguistics*, 2005.

[7] O. Gillet and G. Richard. Transcription and separation of drum signals from polyphonic music. *IEEE Trans. on Audio Speech and Language Processing*, 16(3):529–540, 2008.

[8] D. Huron. Music Information Processing Using the Humdrum Toolkit: Concepts, Examples, and Lessons. *Computer Music Journal*, 26(2):11–26, 2002.

[9] A. P. Klapuri, A. J. Eronen, and J. T. Astola. Analysis of the Meter of Acoustic Musical Signals. *IEEE Trans. on Audio, Speech, and Language Processing*, 14(1):342–355, 2006.

[10] M. Levy. Improving Perceptual Tempo Estimation with Crowd-Sourced Annotations. *Proc. of the 12th Int. Conf. on Music Information Retrieval (ISMIR 2011)*, pages 317–322, 2011.

[11] C. D. Manning and H Schütze. *Foundations of Natural Language Processing*. MIT Press, 1999.

[12] M. Mauch, S. Dixon, C. Harte, M. Casey, and B. Fields. Discovering Chord Idioms through Beatles and Real Book Songs. *Proc. of the 8th Int. Conf. on Music Information Retrieval (ISMIR 2007)*, 2007.

[13] M. Mauch, D. Müllensiefen, S. Dixon, and G. Wiggins. Can Statistical Language Models be Used for the Analysis of Harmonic Progressions? *Proc. of the 10th Int. Conf. on Music Perception and Cognition (ICMPC 2008)*, 2008.

[14] C. McKay. *Automatic Genre Classification of MIDI Recordings*. PhD thesis, McGill University, 2004.

[15] Y. Murakami and M. Miura. Automatic detection system for fill-in from drum patterns employed in popular music. *Proc. of the 10th Western Pacific Acoustics Conference*, 2009.

[16] M. E. J. Newman. Power laws, Pareto distributions and Zipf's law. *Contemporary physics*, 46(5):323–351, 2005.

[17] E. Pampalk. Audio-Based Music Similarity and Retrieval : Combining a Spectral Similarity Model with Information Extracted from Fluctuation Patterns. *Proc. of the 7th Int. Conf. on Music Information Retrieval (ISMIR 2006)*, 2006.

[18] J. K. Paulus and A. P. Klapuri. Conventional and Periodic N-grams in the Transcription of Drum Sequences. *Int. Conf. on Multimedia and Expo (ICME 2003)*, 2003.

[19] G. Peeters. Spectral and Temporal Periodicity Representations of Rhythm for the Automatic Classification of Music Audio Signal. *IEEE Trans. on Audio Speech And Language Processing*, 19(5):1242–1252, 2011.

[20] R. Scholz, V. Dantas, and G. Ramalho. Automating functional harmonic analysis: the Funchal system. *Seventh IEEE International Symposium on Multimedia*, 2005.

[21] G. Sioros and C. Guedes. Complexity-Driven Recombination of MIDI Loops. *Proc. of the 12th Int. Conf. on Music Information Retrieval (ISMIR 2011)*, pages 381–386, 2011.

[22] K. Yoshii, M. Goto, and H. G. Okuno. Automatic Drum Sound Description for Real-World Music Using Template Adaptation and Matching Methods. *Proc. of the 5th Int. Conf. on Music Information Retrieval*, pages 184–191, 2004.

[23] A. Zils, F. Pachet, O. Delerue, and F. Gouyon. Automatic extraction of drum tracks from polyphonic music signals. *Second Int. Conf. on Web Delivering of Music*, pages 179–183. IEEE Comput. Soc, 2002.