ARCHIVES OF ACOUSTICS
33, 1, 43–55 (2008)

n-GRAM-BASED APPROACH TO COMPOSER RECOGNITION

Jacek WOŁKOWICZ, Zbigniew KULKA, Vlado KEŠELJ

Warsaw University of Technology
Institute of Radioelectronics
Nowowiejska 15/19, 00-665 Warszawa, Poland
{j.wolkowicz,z.kulka}@elka.pw.edu.pl

Dalhousie University
Faculty of Computer Science, Canada
e-mail: {jacek,vlado}@cs.dal.ca

(*received June 15, 2007; accepted November 30, 2007*)

This paper describes how tools provided by Natural Language Processing and Information Retrieval can be applied to music. A method of converting complex musical structure to features (n-grams) corresponding with words of text was introduced. Mutual correspondence between both representations was shown by demonstrating certain important regularities known from text processing, which may also be found in music. The problem of automatic composer attribution to which statistical analysis of n-gram profiles known from statistical NLP was applied served as a case study. A MIDI files corpus of piano pieces was chosen as the source of data.

Keywords: composer recognition, music processing, music information retrieval (MIR), n-grams, MIDI.

1. Introduction

Music content processing has become an important domain of research. A lot of work regarding these tasks has already been published. It results from the fact that there are more and more repositories of musical content accessible to everybody through the internet. Meanwhile, tools for searching and browsing the text content, such as Google, have been developed and are widely used. These tools were founded on the basis of IR (Information Retrieval) and NLP (Natural Language Processing). As it has been discovered that people start to create music concurrently with language development, one can assume that music is also a natural language with all repercussions. This implies that some techniques, proven effective within NLP and IR can be applied to music as well. This paper introduces a novel statistical approach to music analysis based on n-grams. The aim of the paper is to show that music is similar to natural languages and can be processed using the methods already developed.

Distinguished levels of text processing, according to JURAFSKY and MARTIN ([10]), are listed in Table 1. NLP, similarly to music processing, tries to cover all levels, from recording to understanding. Of course, there is no such tool that does everything at a time, i.e. understands the meaning and gets knowledge from a raw waveform. In fact, the NLP tools concentrate on a certain level trying to move the problem a level up. Music, similarly to a natural language, can be recorded and presented primarily as a waveform. The "phonetics" level focuses on the investigation of a sound structure, and it deals with notes and instruments separation and identification. This task combined with notes recognition is a problem well known to contemporary sound engineers, even if they do not realize that this gets them involved in the NLP tasks.

Table 1. Levels of NLP. Text vs. music.

	Text processing	Music processing
phonetics	Voice recorded	Recording
phonology	Phonemes of the language	Separated notes
morphology	Words structure	Notes in the score
syntax	Words order	n-grams, notes order
semantics	Words meaning, POS	Harmonic functions
pragmatics	The meaning of a sentence	Phrase structure
discourse	Context of a text	Piece's interpretation

The music content analysis is the next step in so called MLP (Music Language Processing). Music has a hidden structure and hidden rules, much like grammar has for text. It is called the harmony. It rules how to put words (notes) together, how to build phrases with them, which are well-formed. It also manages the musical meaning of the piece which is the order of chords. In the first case, we can talk about the syntax of music while in the second – about the semantics of a certain chord or a pragmatics of a phrase. MIR (Music Information Retrieval) tools should work mainly on those levels. The other problem is that there are no word boundaries in music, and phrasing is driven by harmony, so one has to figure out the structure of the piece as well as its harmonic representation. Related work will be described in Sec. 2. Types of musical data will be introduced in Sec. 3. The method of obtaining n-grams will be presented in Sec. 4. Section 5 contains a description of the dataset used in the experiments. Sections 6 and 7 contains some research done on the dataset. The composer recognition system will be described in Sec. 8 followed by concluding remarks.

2. Related work

2.1. Music Analysis

Music Information Retrieval (MIR) came out of Information Retrieval (IR), the field that is concerned with the structure, analysis, organization, storing, searching and retrieval of relevant information from the large textual databases. Along with the development of multimedia technology, the information content that needs to be made available for searching changed its nature, from pure textual data to multimedia content (text, images, videos and audios). MIR is nowadays a growing international community drawing upon multidisciplinary expertise from computer science, sound engineering, library science, information science, cognitive science, and musicology and music theory [7]. MIR systems, that are operational or in widespread use, have been developed using meta-data such as filenames, titles, textual references and whole non-music information provided with a piece. Now, researches and developers need to face creating content-based MIR systems. The most advanced waveform-like content-based systems rely now upon musical fingerprint idea. It insists on creating a small set of features that may be simply extracted from the piece and retrieving information based on these features [14].

The most important research area in this case is work done at the field of symbolic music representation. With pitch and rhythm dimensions quite easily obtainable from music data, one can obtain the textual string representation of the music and then try to apply text based techniques to solve MIR tasks. The main problem is to define the relation between pitch with rhythm information and musical text representation.

Various music representations have been already proposed. BUZZANCA [4] proposed using symbolic notes meanings, i.e. pitches like c', d', c" and durations like 'quarter-note', 'half-note' instead of using absolute values for pitch and duration. However, the task, which was taken, was classification of highly prepared themes representing the same type of music. Moreover, these features were given then as an input to a neural network, so one does not know what was really taken into consideration. This is the main drawback of neural networks, because we do not have any feedback from the network whether our ideas and assumptions are valid or not. THOM ([19, 20]) suggests splitting the piece on bars. She contends that using fixed length, gliding window could make the problem sparse. It is true, however, as the research conducted in this work show, that modern computers could successfully handle even such a sparse problem. The next example is the Essen Folksong Collection. It provides a large sample of mostly European folksongs that have been collected and encoded under the supervision of Helmut Schaffrath at the University of Essen (see [16–18]). Each of the 6,251 folksongs in the Essen Folksong Collection is annotated with the Essen Associative Code (ESAC) which includes pitch and duration information ([2, 3]). In this approach the pitch is given explicitly, while regarding time, we can say, that this information is more flexible because it gives us the information about relative duration of the first (or shortest) note in the passage. Another approach was presented in [8]. They use original MIDI pitch representation and absolute time value with 20 ms resolution.

Unlike all the approaches presented above, MIR researchers prefer approaches similar to the one presented in this work. The first such approach was introduced by DOWNIE [6]. In this work, only a pitch was encoded as an interval between two consecutive notes. A more precise approach was presented by DORAISAMY [5]. She encoded both pitch (as an interval to the previous note) and duration ratio (as a ratio of durations of 2 consecutive notes). However, she did not apply logarithm transformation to it. In the work regarding theme classification provided by POLLASTRI and SIMONCELLI [15] an approach to take relative pitch and relative duration was also used. However, they quantified both dimensions that they obtain 3 different values for time and 5 – for pitch.

2.2. Composer Recognition

A system that was successfully applied to the problem of authorship attribution on texts has been published by KEŠELJ, PENG, CERCONE and THOMAS [11]. They reported that a successful authorship attribution method can be applied to text using n-gram based statistical approach from natural language processing with the accuracy that reaches 100%. The method introduced is very simple in its concepts and might be successfully applied in other fields like music.

POLLASTRI and SIMONCELLI [15] have developed a system of theme recognition using Hidden Markov Model and report 42% accuracy among 5 composers. This is not a satisfactory performance. However, they claimed, according to other psychological research, that human ability of recognizing themes for professionals is about 40%. They have also used n-grams, as it was described in the previous section and they have done their research just on monophonic themes.

A successful style recognition system has been done by BUZZANCA [4]. He used neuronets and reports 97% accuracy, but "highly prepared data" were used in this solution. By "highly prepared data", one means selecting themes from pieces, not giving whole pieces to be classified. Having that in mind, this solution is not fully automated, because it involves long-lasting users, experts work on data preprocessing, which is not the case in this thesis. Second, the use of neuronets cannot give an explanation of such behavior and results. It does not give the insight into the features that can distinguish between different composers. The system may work, but it will not increase human knowledge in this area. In the n-gram-based approach one assumes, that the order of notes plays role and after that one can take the profiles out and check what the features (the sequences of notes) that specify composer's contribution are.

Lots of work has been done to recognize some aspects of waveform data using different methods ([1, 9, 13]), but this field is so far not investigated enough and the results are quite poor. The main problem in this field is that we still cannot interpret the waveform data well and without this insight our work is still just a rambling in the darkness.

3. Types of musical data

There are two, quite different types of musical data, that can be stored on computers.

1. Raw – the recorded sound, compressed (e.g. mp3 format) or stored as PCM files (e.g wav format).
2. Symbolic representation – score notations, mus (Finale), sib (Sibelius), abc (abc-Music notation), xml (Music XML), or finally the MIDI protocol.

People got used to raw representations because they like to hear "real" artists' music, not a symbolic version, which is played on every machine differently. The other reason of this situation comes from the fact that not everyone understands music in the way one reads text. Musical education and studying scores are common in the society.

Same happens with text: on the one hand we may store the original author's voice, on the other – textual data. Unlike music, people prefer to represent text symbolically. This representation is easy to store, edit (using text editors and keyboards), and process. It is also well known to most people from their childhood. The second issue is the flatness of the text – words occur one after another, there is no concurrency in the text. It is not so simple in music, and this fact should be resolved before applying NLP tools to music. The next thing is the fact that almost everyone can read. That is why people prefer raw music formats and symbolic texts formats.

MIDI files store symbolic data and they may behave like textual files. Nevertheless, they consist of concurrent channels and tracks, which may overlap and on each channel notes may also co-occur. The resulting output is much like a crowd play where lots of people talk at the same time. Thus, when talking about correspondence to text, one has to omit these concurrencies. We decided to treat channels separately, solving the problem of parallelism in each channel independently by removing notes that co-occur. In each channel, we took the highest currently played note because, according to basic psychoacoustic knowledge, it is assumed that people concentrate on them [21].

4. Musical data representation – n-grams extraction

n-gram is simply n consecutive letters or words. There are "word" and "character" n-grams. They overlap, i.e. each token belongs to "n" n-grams. For instance, in text "Music" we have 3 character 3-grams: "Mus", "usi" and "sic". n-grams are very useful in NLP in the situations, where not only words are significant, e.g. in authorship attribution, language recognition or where it is hard to separate the words. A good example is Thai, where there are no whitespaces. In this aspect, Thai is especially similar to music – for us it is just a flow of characters without an order or semantics; however it still remains a natural language for Thais. If the NLP tools may also be applied to this language, why they cannot be applied to music as well (treated as a natural language).

The first step of n-gram extraction after simplifying the data from MIDI files (i.e. making linear order of notes in each track), is to find what could represent unigrams. The simplest approach would be just getting the duration or pitch as the basic feature, but this does not bring good results. The pieces can be played at different speeds and can be transposed to any key. The features one needs have to be key independent so that not the absolute note pitch is important, but the relative pitch to other notes. It is crucial, because the key does not tell us anything about a certain work, e.g. J. S. Bach wrote two sets of preludes and fugues, each fugue in each existing key in well tempered scale, thus if one does the pitch distribution analysis, we will obtain one flat, normalized. The second important feature of musical n-grams is that they should be tempo-independent. In MIDI files the duration is not given symbolically – as quarters, eights, half-notes, but in a direct way, that can be mapped to milliseconds. Every MIDI file representing the same piece, but sequenced by different people (or programs) will look a little bit different. That is why we decided to apply a relative duration counting, not the direct one. Each difference is in a logarithmic scale and is quantified to cover some random tempo fluctuations. Quantization step of 0.2 was applied, i.e. 0, 0.2, 0.4... . The formula applied to each pair of notes is given as follows:

$$(P_i, T_i) = \left(p_{i+1} - p_i, \text{round}\left(\log_2\left(\frac{t_{i+1}}{t_i}\right)\right)\right), \tag{1}$$

where p_i denotes the i-th note pitch (in MIDI units), t_i stands for the i-th note length (in ms) and (P_i, T_i) is the resulting tuple. The procedure of extracting n-grams is shown in Fig. 1.

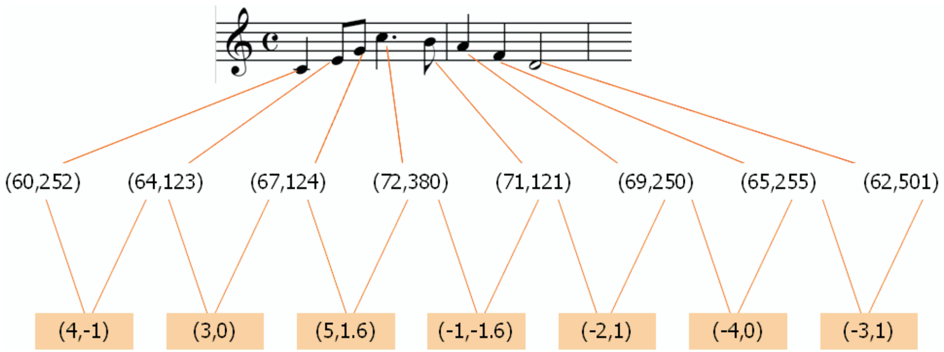


Fig. 1. Unigrams extraction.

A transition from uni-grams to n-grams is simple and means getting n consecutive unigrams as an item. Three types of n-grams can be obtained out of this. We can consider the rhythm only, or the melody only, and we can also take n-grams as a combination of both these features. The n-gram representation is quite similar in its form to text. We claim that MIR engines may be built using this representation, working according to the same rule as IR engines like Google. The great work about string matching techniques, with a result of a MIR system was shown in LEMSTROM's dissertation ([12]). The tool is available online, but its methods still need enhancements.

5. MIDI corpus

We collected a set of MIDI files freely available on the Internet of five different composers, and chose only the piano works for better compatibility. Moreover, each piece had to be well-sequenced, i.e. each channel had to represent one and only one staff or hand. The reason for this is that it is very easy to produce a MIDI sequence that sounds well, but is messy inside. The number of pieces and their size are given in Table 2:

Table 2. MIDI corpus properties.

	Composer	Training Set	Testing Set
1	J. S. Bach	99 items, 890 kB	10 items, 73 kB
2	L. van Beethoven	34 items, 1029 kB	10 items, 370 kB
3	F. Chopin	48 items, 870 kB	10 items, 182 kB
4	W. A. Mozart	15 items, 357 kB	2 items, 91 kB
5	F. Schubert	18 items, 863 kB	5 items, 253 kB

While considering music files, it is necessary to point out, that there are big disproportions between pieces. Some miniatures are quite tiny, but there are also very large forms, like concertos. Thus, it is better to describe the volume of corpora in bytes rather than in the number of pieces. The second important issue is to know that differences between composers come from composers' background and their lifetimes, e.g. a greater difference is between F. Schubert and J. S. Bach than between F. Schubert and F. Chopin.

6. Zipf's law for music

There is a certain number of regularities and laws that form the basis of NLP and IR. These laws show that text is not a set of words distributed arbitrarily. Below we present a very important law – Zipf's law, which describes the distribution of words in text [23]. It allows estimating certain features of an IR system before implementing and running it.

Figure 2 was obtained for the piano pieces of the corpus described above. It shows the number of each n-gram occurrences as a function of its rank, i.e. the place of each "word" in the sorted frequency table in descending order. According to Zipf's law, the frequency of any word is roughly inversely proportional to its rank. If dimensions are in a logarithmic scale, the relation should be linear.

Despite some irregularities at the beginning, the law is satisfied. We may notice the difference between rhythmic and melodic profiles. There are much more low-rank rhythmic n-grams than melodic ones and much more high-rank melodic n-grams than rhythmic ones. It means that rhythm is usually much simpler than melody and that melody remains unique for every theme because most of melodic patterns occur few times.

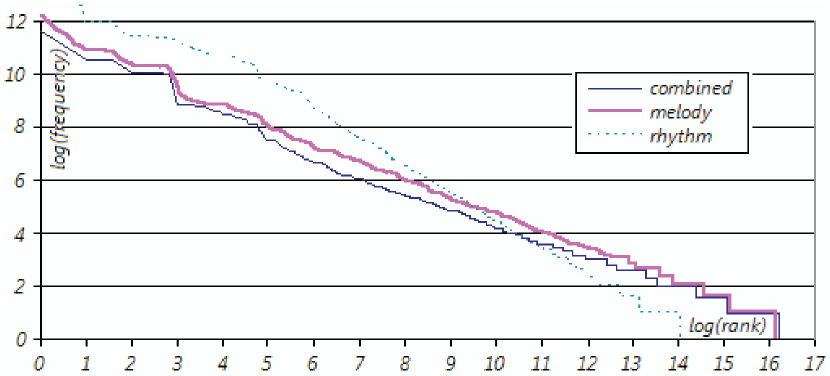


Fig. 2. Zipf's Law for music for three types of n-grams.

7. Entropy analysis

We can distinguish certain groups of words in text, such as key-words, stop-words, noise-words. Key-words are the words with a meaning and important semantic value for the text. Their rank should be in the middle of a logarithmic rank value. Stop-words are the most frequent words, like “the”, “a”, “and”. They do not have any semantic meaning and usually mess up analysis. Noise-words are the words that occur few times and they do not make us come to any conclusion. The definitions of the preceding groups of words are “semantic” so they cannot be applied to music, i.e. we cannot simply name a phrase a key-word. The notion that may help in this situation is entropy as a measure of information.

The feature that is a good indicator between classes needs to occur quite frequently in all the documents belonging to the certain class (i.e. the entropy of this term in a certain class should be high), but has to be quite rare in the documents that do not belong to the class. (i.e. The entropy of entropies calculated in each class should be as small as possible). Thus, (whereas the maximum entropy on N elements is equal $\log_2 N$) the rank of each term denoted as:

$$R(i) = \max_{k \in 1..N} (H(i,k)) \left(\log_2 N - \underset{k \in 1..N}{H} (H(i,k)) \right) \tag{2}$$

should be large if the term is a good discriminator between certain classes, and low, if it does not discriminate well. Regarding entropy, key-words are the words (n-grams) with high entropy inside the class, and small entropy among all classes. Hence, noise-words are the n-grams with high entropy in both areas while stop-words have both rates low.

The limiting value for being a key word is the value of $\log_2 N$ where N is the number of classes. In this case there are only two occurrences of the term and they dropped luckily into the same class. The probability of this event is $1/N$ so a random classifier will obtain the same accuracy. Listing all the terms sorted by R in descending order shows the following groups with:

1. $R(i) > \log_2 N$ (‘key words’),
2. $R(i) = \log_2 N$ (‘random pairs’),

3. $R(i) < \log_2 N$ ('stop words'),
4. $R(i) = 0$ ('noise words').

The first group contains the words that bring us most information about its classes. Second one is the random pairs group described above. Terms from the third group bring us less information than random words; these are stop words, that occur equally frequently in every group. The fourth group (noise words) represents words that occur usually at most one time in every group.

After counting all occurrences in each group we obtained a distribution shown in Fig. 3 (on the vertical axis there is the proportion of each group, on the horizontal – the log rank of each term assigned during calculating Zipf's law). More details of the method may be found in this co-author's dissertation ([3]). One may notice, that the position of each group is as expected, so the structure of music pieces corresponds to the one of text documents, which shows, that music can be treated as a natural language.

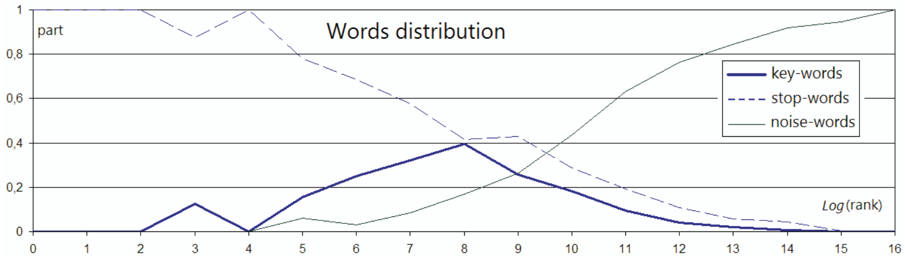


Fig. 3. n-grams' distribution among corpus.

8. Composer recognition task as an example of n-gram-based approach to music analysis

According to conclusions derived from previous sections, the NLP tools used already for text may also be applied to music. When it was shown that the tools can be successfully applied to text to solve the issue of authorship attribution [11], we decided to investigate the composer recognition task using them. However, the use of this method in the research of musical content is a novel approach.

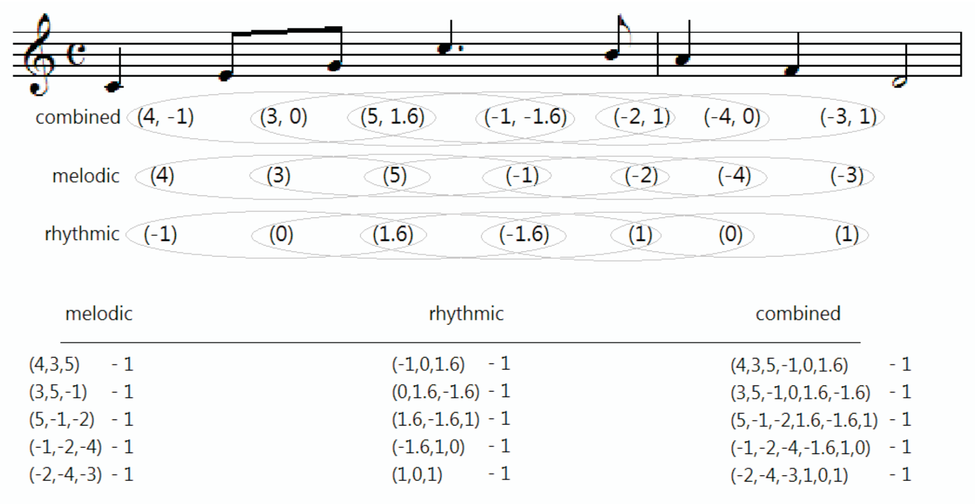
Similarly as in the authorship attribution, we created a profile of each composer as a table containing n-grams with their occurrences in all pieces of each composer from the training corpus. If a new piece comes to the system, the program counts all occurrences of each n-gram and creates a profile of the testing piece. The profile is then compared to composers' profiles and the most similar is taken as a result. The details, how the profiles are being build and the other details of the algorithm will be given in the following sections.

8.1. Algorithm details

According to the fact, that each uni-gram contains two values, that represent pitch and rhythm change, one can create three types of n-grams:

1. melodic – if only pitch information is taken into consideration,
2. rhythmic – if only rhythm information is taken into consideration,
3. combined – where both, melodic and rhythmic factors, create features.

And according to this fact three types of profiles can be obtained out of these n-grams types (see Fig. 4):



melodic		rhythmic		combined	
(4,3,5)	- 1	(-1,0,1.6)	- 1	(4,3,5,-1,0,1.6)	- 1
(3,5,-1)	- 1	(0,1.6,-1.6)	- 1	(3,5,-1,0,1.6,-1.6)	- 1
(5,-1,-2)	- 1	(1.6,-1.6,1)	- 1	(5,-1,-2,1.6,-1.6,1)	- 1
(-1,-2,-4)	- 1	(-1.6,1,0)	- 1	(-1,-2,-4,-1.6,1,0)	- 1
(-2,-4,-3)	- 1	(1,0,1)	- 1	(-2,-4,-3,1,0,1)	- 1

Fig. 4. Building profiles from a tune.

Trigrams are used in the example. In this case each n-gram occurs once. However, as far as whole pieces are concerned, some n-grams are more frequent than others. Each profile is a table containing n-grams as keys and numbers of occurrences as values. As a result, one obtains three independent profiles and they are analyzed in next steps separately.

The next step of the algorithm lies in creating the profiles for the analyzed piece. This part works similarly to the previous one – composers' profiles part. Here, the piece that is being recognized is converted to the same form as original profiles – each piece is also represented as three vectors of n-grams occurrences of each type. These vectors are then compared with the appropriate profiles of composers using the following similarity measure (it is a modified method described by KEŠELJ, PENG, CERCONE and THOMAS [11] that was used for comparing the profiles of texts authors):

$$\mathrm{Sim}\left(\mathbf{x},\mathbf{y}\right)=\sum\left(4-\left(\frac{2\cdot(x_i-y_i)}{x_i+y_i}\right)^2\right), \tag{3}$$

where **x** and **y** stands for a profile (any type) of a composer and corresponding profile of a piece. One obtains $3n$ similarity values out of these calculations, where n stands for the number of analyzed composers and 3 comes from the number of profile's types. There are many possible judgment algorithms that can be applied in order to find the most appropriate choice. Thus it is not connected with composer classification but classification itself, therefore a decision not to pay an attention on this parameter was made. The following steps were applied:

1. Sum up all the similarities for profiles of each composer.
2. Sort all sums descending.
3. Take a composer with a highest sum as a result.

Sample judgment calculations are shown in Table 3. For details of the algorithm, please look in the co-author's dissertation [22].

Table 3. Evaluation of the Frederic Chopin prelude Op. 28 No. 22.

		Profiles			Total	Verdict
		melodic	rhythmic	combined		
Composers	Beethoven	43.2	17.2	11.0	71	3
	Mozart	49.2	11.4	6.4	67	4
	Bach	62.4	8.2	6.4	77	2
	Schubert	19.3	13.2	5.9	38	5
	Chopin	86.8	25.1	10.9	122	**1**

8.2. *Results*

There are certain degrees of freedom in the system. The algorithm was tested for different n-gram lengths (n), profile sizes and aging factor at the time of composers' profiles creation. The best results were obtained for the aging factor (0.96) and they are shown in Table 4 (with varying n-grams lengths and profiles sizes). Accuracy, i.e. the ratio of correctly assigned pieces to the total number of pieces in the test collection, reaches 84% for the highest profile sizes and for $n = 6$. It might mean, that the average "music" word has usually about 7 notes (6-gram describes 7 consecutive notes), which is a (musical) measure (notes between two consecutive bars). The important thing to be pointed out is that the random classifier can obtain the accuracy of 20%, so the result over 80% is a good result. The second thing is the fact, that some pieces were written by

Table 4. Results of the algorithm.

size n	100	250	500	1000	2500	5000	10000
2	0.41	0.38	0.38	0.35	0.32	0.43	0.43
3	0.46	0.54	0.59	0.62	0.59	0.51	0.43
4	0.62	0.70	0.65	0.73	0.73	0.78	0.86
5	0.54	0.62	0.70	0.78	0.78	0.81	0.81
6	0.54	0.59	0.68	0.68	0.84	0.78	0.84
7	0.46	0.49	0.68	0.68	0.68	0.70	0.84
9	0.46	0.57	0.49	0.51	0.57	0.68	0.76
12	0.41	0.46	0.41	0.41	0.41	0.46	0.49

a composer in a different style and it is really hard even for people, who do not know the certain piece, to classify the piece to the proper class. That's why the algorithm might not reach 100%.

9. Conclusions

Our analysis shows that music can be processed by NLP and IR tools and some aspects of this problem were proved in this paper. Showing that some of the methods from natural language processing work on music lead us to the point where we can try to populate other methods, such as clustering, plagiarism detection, music information retrieval systems and much more. Using the n-gram interpretation may allow to index and efficiently browse musical libraries, which is a major problem nowadays. The usefulness of the methods was proven in the case of composer recognition; however, we claim that there are plenty of tasks that may be solved using these methods.

References

[1] ALLAMANCHE E., HERRE J., HELLMUTH O., FRÖBA B., KASTNER T., CREMER M. , *Content-based identification of audio material using MPEG-7 low level description*, In Proceedings of the International Symposium of Music, 2001.

[2] BOD R., *Probabilistic grammars for music*, Proceedings of the Belgian-Dutch Conference on Artificial Intelligence, 2001.

[3] BOD R., *A unified model of structural organization in language and music*, Journal of Artificial Intelligence Research, **17**, 289–308 (2002).

[4] BUZZANCA G., *A supervised learning approach to musical style recognition*, Proceedings of International Computer Music Conference, 1997.

[5] DORAISAMY S., *Polyphonic music retrieval: The n-gram approach*, Ph.D. thesis. University od London, 2004.

[6] DOWNIE S., *Evaluating a simple approach to music information retrieval: Conceiving melodic n-grams as text*, Ph.D. Thesis, University of Western Ontario, 1999.

[7] DOWNIE S., *Music information retrieval*, Annual Review of Information Science and Technology 37, 295–340 (2003).

[8] FRANCU C., NEVILL–MANNING C. G., *Distance metrics and indexing strategies for a digital library of popular music*, IEEE International Conference on Multimedia and Expo (II), 2000.

[9] FRANKLIN D. R, CHICHARO J. F., *Paganini – a music analysis and recognition program*, Fifth International Symposium on Signal Processing and its Applications, Brisbane, **1**, 107–110 (1999).

[10] JURAFSKY D., MARTIN J. H., *Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition*, 1st Ed. Prentice Hall PTR. ISBN 0130950696, 2000.

[11] KEŠELJ V., PENG F., CERCONE N., THOMAS C., *N-gram-based author profiles for authorship attribution*, Proceedings of the Conference Pacific Association for Computational Linguistics, PACLING'03, 255–264 (2003).

[12] LEMSTROM K., *String matching techniques for music retrieval*, Ph.D. Thesis, University of Helsinki, Finland, 2000.

[13] MARTIN. K. D., *Sound-source recognition: A theory and computational model*, Ph.D. Thesis, Massachusetts Institute of Technology, 1999.

[14] PARDO B., *Finding structure in audio for music information retrieval*, IEEE Signal Processing Magazine, **23**, 4, 126–132 (2006).

[15] POLLASTRI E., SIMONCELLI G., *Classification of melodies by composer with hidden Markov models*, Proceedings of the First International Conference on Web Delivering of Music, pp. 88–95, 2001.

[16] SCHAFFRATH H., *Repräsentation einstimmiger Melodien: computerunterstützte Analyse und Musikdatenbanken*, Enders B., Hanheide S. [Eds.], Neue Musiktechnologie, pp. 277–300, B. Schott's Söhne, Mainz 1993.

[17] SCHAFFRATH H., HURON D. [Ed.], *The Essen folksong collection in the humdrum kern format*, CCARH, Menlo Park, California 1995.

[18] SELFRIDGE–FIELD E., *The Essen musical data package*, CCARH, Menlo Park, California 1995.

[19] THOM, B., *Unsupervised learning and interactive Jazz/Blues improvisation*, Proceedings of the Seventeenth National Conference on Artificial Intelligence, pp. 652–657, 2000.

[20] THOM B., *BoB: an interactive improvisational music companion*, Proceedings of the Fourth International Conference on Autonomous Agents (Agents-2000), Barcelona, Spain, 2000.

[21] UITDENBOGERD A., ZOBEL J., *Melodic matching techniques for large database*, Proceedings of the seventh ACM international conference on Multimedia, pp. 57-66, 1999.

[22] WOŁKOWICZ J., *N-gram-based approach to composer recognition*, M.Sc. Thesis, Warsaw University of Technology, 2007, www.cs.dal.ca/∼jacek/papers/thesis.pdf

[23] ZIPF G. K., *Human behavior and the principle of least effort: An introduction to human ecology*, Addison-Wesley Press, Cambridge, 1949, ISBN 978-0-028-55830-1.