TEXT INFORMATION RETRIEVAL APPROACH TO MUSIC INFORMATION RETRIEVAL

Jacek Wolkowicz
Faculty of Computer Science
Dalhousie University
jacek@cs.dal.ca

Vlado Kešelj
Faculty of Computer Science
Dalhousie University
vlado@cs.dal.ca

ABSTRACT

This MIREX submission for symbolic music similarity task adopts textual information retrieval methodology in the process of music information retrieval. The main contribution of this approach is to utilize well established term weighting methods for text retrieval and check their suitability for music data. We use a simple feature extraction method, so that the performance of an algorithm depends only on the applied term weighting function. The parameters for each of the algorithms are optimized based on 2005 SMS MIREX data.

1. INTRODUCTION

Submissions using methods derived from Textual Information Retrieval have been seen in the previous editions of MIREX SMS (Symbolic Melodic Similarity) task [4–6]. The methods they used are well suited to textual representations if we deal with documents using a 'bag-of-words' approach. The same can be done with music data, especially if one does not have to deal with concurrencies, i.e. monophonic music. The main focus is paid to how to transfer the input sequence of notes into a set of features and how to compare (measure the similarity) between different feature sets.

In this submission we will try to evaluate a different aspect of music document retrieval. Using a simple term extraction method, we will focus on those parts of IR process (and finding similar documents to the query could be seen as a document retrieval process), that are usually in focus of text IR research. Our goal is to evaluate how different term weighting functions affect retrieval performance. In textual IR it is important to determine which terms are more important for each document or the dataset as a whole. Some frequent terms (dubbed stopwords) don't usually even take part in the retrieval process at all. This reduces retrieval time, but also allows for better ordering of the retrieval results, which is also the main point of MIREX SMS task. We will adjust the parameters of the retrieval system based on the analysis of the previous 2005 MIREX data.

2. METHODOLOGY

The process of document retrieval starts with extracting features from the corpus documents. Input dataset consists a set of standard MIDI files, each containing a single track of notes representing a monophonic melody. Since none of the notes are concurrent or overlap, string based methods can be directly applied to the input documents [1, 3, 10]. Like text documents that can be just seen as series of characters, monophonic music opi are just series of notes. The difference with music files is that text documents are easily separable into basic features — words. Since there is no such a thing as a clear phrase boundary in music, the usual bag-of-words, or bag-of-terms approach consists of building n-grams, i.e. substrings of n consecutive tokens (notes) that start with every note.

Each of the features that conforms an n-gram, which we call it here, a uni-gram, is derived from each note event that one finds in the notes stream. It can either contain absolute values representing music features, such as note's pitch, duration or inter-onset interval (IOI), but in most cases relative (interval) features are used. We went for the last approach using either melodic intervals or a combinations of melodic and inter-onset interval ratios (IORs). Those features satisfy pitch and tempo invariance, which is one of the basic requirements in music information retrieval systems. The other question is how one should deal with the numbers that represent melodic intervals and IORs, i.e. at which level of granularity they should be dealt with. To answer this question, an analysis have been conducted based on the already released previous MIREX data. We have found that the more precise the features, the better the results one can obtain. For our submissions we used either pure melodic intervals or melodic intervals combined with binary logarithms of IOR's (inter-onset interval ratios) indicating how

much the duration of a note have changed.

With our tests we were also able to determine the optimal n for each of the proposed algorithms. It varies from 2 to 5, depending on the parameters. The general rule of thumb is though, the more general the terms (features), the bigger the n should be. We have also tested various similarity measures figuring out that a simple cosine similarity, widely used in textual information retrieval, gives good results with music data. The basic formula for the cosine similarity is the following:

$$sim(\mathbf{x}, \mathbf{y}) = \frac{\sum_i x_i y_i}{\sqrt{\sum_i x_i^2}\sqrt{\sum_i y_i^2}} \quad (1)$$

where x_i is a weight of a term i in a document $\mathbf{x}$. The main contribution of this submission is how different, text-based term weighting measures affect music information retrieval.

The parameters of proposed algorithm have been tuned based on previous, MIREX 2005 dataset consisting of a subset of RISM collection [8, 9]. Based on that and using average dynamic recall (ADR) [7] as a performance measure, we were able to evaluate the best parameters for each of the proposed approaches and draw general conclusions about the behaviour of the systems.

3. ALGORITHMS

3.1 Binary weights (WK1)

The WK1 algorithm uses a basic binary term weighting approach. It is either 0 (if a term, or an n-gram doesn't appear in the document) or 1 (if a term appears in the document), so the resulting similarity measure between two documents is the number of terms in common normalized by geometric average of numbers of unique terms in both documents. This algorithm got the best results on 2005 SMS MIREX dataset for melodic intervals used as features and n being 5.

3.2 Term count weights (WK2 and WK3)

The following two algorithms use simple term counts (the number of times the term appears in the compared documents) which gives a classical cosine similarity definition. This rather simple method gave us surprisingly good results for two different settings so we have decided to submit both for the competition. The first one (WK2) uses again melodic intervals as simple features and $n = 4$, while the second one uses a combinations of melodic and IOI intervals with $n = 2$. The relative performance of these two algorithms will allow to assess whether introduction of rhythmic features helps to improve the overall score.

3.3 tf.idf term weights (WK4)

WK4 algorithm computes standard tf.idf weights of each term from documents to compare, which gives each term i a weight depending on its count (c_i) within the document (d) and in how many documents of the collection (D) a term i occurs. The formula is given as follows:

$$tf.idf_i = \frac{c_i}{\|d\|} \log \frac{\|D\|}{\delta_i} \quad (2)$$

where $\delta_i = \|\{d \in D | i \in d\}\|$ is the number of documents containing term i. This measure is commonly used in Textual Information Retrieval for term weighting so it would be interesting, how it performs in music challenge. For the settings of WK4 we have again determined, that n equals 4 and melodic interval features worked the best in our tests.

3.4 Okapi BM25 (WK5, WK6)

BM25, unlike tf.idf, is an industry-developed weighting scheme, that typically outperforms classic term weighting measures, like tf.idf. It tries to capture roughly the same concept as original tf.idf measure but tries to balance documents with different lengths and different term distribution:

$$bm25_i = \frac{c_i(k+1)}{c_i + k(1 - b + b\frac{\|D\|}{avgdl})} \log \frac{\|D\| - \delta_i + 0.5}{\|D\| + \delta_i} \quad (3)$$

where $avgdl$ is an average document length. It is parametrized, with parameters b and k, and we have used a recommended setting of $b = 0.75$ and $k = 2$. Since it should be a top performing function, we have came up with two sets of settings: WK5 with melodic interval features of length 4 and WK6 with features combining melodic interval and IOI ratios with n-gram length of 2.

4. RESULTS

Our algorithms were evaluated along with 5 other submitted algorithms reaching similar total score. Only the UL series algorithms outperformed most of our submissions, yet still the difference in most cases was not measured as significant (apart from UL1) [2]. For most measures, only cumulative results were published, which does not allow us to draw many conclusions about the actual algorithm performance, however for the purpose of performing Friedman test with multiple comparison results, a FP10 results for each query and each algorithm have been published. FP10 stands for fine precision at 10 and is the sum of all the fine ratings of all the items in each of the result sets. The results for each query type are collected in the Table 1. The best and the worst performers for each query have been highlighted and all the values — colour coded for clarity. According to them, UL3 scored the best in all 5 categories and was the best algorithm for this measure overall. UL1 and one of our submissions, WK1 came on paar second. However, what drew our attention were the fine results calculated for

Table 1. Results of SMS 2011 task calculated for each query modification type separately. The numbers represent FP10 values, in percents.

	LJY1	LJY2	UL2	UL1	UL3	WK1	WK2	WK3	WK4	WK5	WK6
overall	**64**	**66**	**65**	**68**	**73**	**68**	**67**	**66**	**65**	**65**	**65**
no errors	67	69	68	70	72	68	69	67	68	69	67
deleted	67	68	62	69	76	70	67	65	64	65	62
inserted	59	61	66	66	70	65	64	64	63	62	65
enlarged	64	65	64	69	73	67	66	65	63	64	65
compressed	65	67	65	63	72	67	67	67	67	67	67

Table 2. Results of SMS 2011 task calculated for each base query separately. The numbers represent FP10 values, in percents.

	LJY1	LJY2	UL2	UL1	UL3	WK1	WK2	WK3	WK4	WK5	WK6
q01	42	48	62	**64**	44	47	50	44	48	47	52
q02	69	69	73	70	71	72	56	42	44	44	24
q03	43	51	56	62	50	68	62	31	56	62	46
q04	50	48	56	57	61	56	41	53	48	44	61
q05	45	46	56	64	64	52	56	48	53	56	70
q06	49	49	58	57	45	36	41	53	52	40	39

each query separately (see Table 2). The table consists of FP10 values achieved by each of the algorithms for each base query only, which in essence breaks down the 'no errors' row from Table 1. It turned out that a lot depends on the actual query, since our most sophisticated setup, WK6 although it performed rather poorly overall (it was one of the algorithms that came last in this category), achieved the best scores in two out of six queries. Since one knows nothing about the actual queries, because this also is kept confidential at MIREX, it does not allow to draw any meaningful conclusion why it happened, but one can clearly see that the type of the actual query should also play an important role in determining the best algorithm for the task.

5. CONCLUSIONS

We have analyzed how different term weighting methods influence the performance of n-gram-based similarity measures. Our algorithms used simple and well established concepts from textual information retrieval, and yet, came close to the leaders of the competition. We have also shown a big variability of the results depending on the query selection process which indicate that the problem of finding the best music melodic similarity measure is not yet over, as there are more factors that have to be taken into consideration that we have not looked into. More information about the actual queries used in 2011 MIREX SMS task would help to target those factors.

6. REFERENCES

[1] S. Doraisamy. *Polyphonic Music Retrieval: The N-gram Approach*. PhD thesis, University of London, 2004.

[2] International Music Information Retrieval Systems Evaluation Laboratory. Mirex 2011 challenge on symbolic melodic similarity, August 2011.

[3] K. Lemstrom. *String matching Techniques for Music Retrieval*. PhD thesis, University of Helsinki, Helsinki, Finland, 2000.

[4] Nicola Orio. Combining multilevel and multi-feature representation to compute melodic similarity. In *MIREX*, 2005.

[5] Iman S. H. Suyoto and Alexandra L. Uitdenbogerd. Simple efficient n-gram indexing for effective melody retrieval. In *MIREX*, 2005.

[6] Iman S. H. Suyoto and Alexandra L. Uitdenbogerd. Simple orthogonal pitch matching with ioi symbolic music matching. In *MIREX*, 2010.

[7] R. Typke, R.C. Veltkamp, and F. Wiering. A measure for evaluating retrieval techniques based on partially ordered ground truth lists. In *Multimedia and Expo, 2006 IEEE International Conference on*, pages 1793–1796, july 2006.

[8] Rainer Typke, Marc den Hoed, Justin de Nooijer, Frans Wiering, and Remco C. Veltkamp. A ground truth for half a million musical incipits. *Journal of Digital Information Management*, 3:34–39, 2005.

[9] Julián Urbano, Mónica Marrero, Diego Martín, and Juan Lloréns. Improving the Generation of Ground Truths based on Partially Ordered Lists. In *International Society for Music Information Retrieval Conference*, pages 285–290, 2010.

[10] Jacek Michał Wołkowicz. N-gram-based approach to composer recognition. Master's thesis, Warsaw University of Technology, Warsaw, Poland, 2007. Supervisor-Kulka Zbigniew.