CADENCE DETECTION IN WESTERN TRADITIONAL STANZAIC SONGS USING MELODIC AND TEXTUAL FEATURES

Peter van Kranenburg, Folgert Karsdorp
Meertens Institute, Amsterdam, Netherlands
{peter.van.kranenburg,folgert.karsdorp}@meertens.knaw.nl

ABSTRACT

Many Western songs are hierarchically structured in stanzas and phrases. The melody of the song is repeated for each stanza, while the lyrics vary. Each stanza is subdivided into phrases. It is to be expected that melodic and textual formulas at the end of the phrases offer intrinsic clues of closure to a listener or singer. In the current paper we aim at a method to detect such cadences in symbolically encoded folk songs. We take a trigram approach in which we classify trigrams of notes and pitches as cadential or as non-cadential. We use pitch, contour, rhythmic, textual, and contextual features, and a group of features based on the conditions of closure as stated by Narmour [11]. We employ a random forest classification algorithm. The precision of the classifier is considerably improved by taking the class labels of adjacent trigrams into account. An ablation study shows that none of the kinds of features is sufficient to account for good classification, while some of the groups perform moderately well on their own.

1. INTRODUCTION

This paper presents both a method to detect cadences in Western folk-songs, particularly in folk songs from Dutch oral tradition, and a study to the importance of various musical parameters for cadence detection.

There are various reasons to focus specifically on cadence patterns. The concept of cadence has played a major role in the study of Western folk songs. In several of the most important folks song classification systems, cadence tones are among the primary features that are used to put the melodies into a linear ordering. In one of the earliest classification systems, devised by Ilmari Krohn [10], melodies are firstly ordered according to the number of phrases, and secondly according to the sequence of cadence tones. This method was adapted for Hungarian melodies by Bártok and Kodály [16], and later on for German folk songs by Suppan and Stief [17] in their monumental *Melodietypen des Deutschen Volksgesanges*. Bronson [3] introduced a number of features for the study of Anglo-American folk song melodies, of which final cadence and mid-cadence are the most prominent ones. One of the underlying assumptions is that the sequence of cadence tones is relatively stable in the process of oral transmission. Thus, variants of the same melody are expected to end up near to each other in the resulting ordering.

From a cognitive point of view, the perception of closure is of fundamental importance for a listener or singer to understand a melody. In terms of expectation [8, 11], a final cadence implies no continuation at all. It is to be expected that specific features of the songs that are related to closure show different values for cadential patterns as compared to non-cadential patterns. We include a subset of features that are based on the conditions of closure as stated by Narmour [11, p.11].

Cadence detection is related to the problem of segmentation, which is relevant for Music Information Retrieval [21]. Most segmentation methods for symbolically represented melodies are either based on pre-defined rules [4, 18] or on statistical learning [1,9,12]. In the current paper, we focus on the musical properties of cadence formulas rather than on the task of segmentation as such.

Taking Dutch folk songs as case study, we investigate whether it is possible to derive a general model of the melodic patterns or formulas that specifically indicate melodic cadences using both melodic and textual features. To address this question, we take a computational approach by employing a random forest classifier (Sections 5 and 6).

To investigate which musical parameters are of importance for cadence detection, we perform an ablation study in which we subsequently remove certain types of features in order to evaluate the importance of the various kinds of features (Section 7).

2. DATA

We perform all our experiments on the folk song collection from the Meertens Tune Collections (MTC-FS, version 1.0), which is a set of 4,120 symbolically encoded Dutch folk songs. [1] Roughly half of it consists of transcriptions from field recordings that were made in the Netherlands during the 20th century. The other half is taken from song books that contain repertoire that is directly related to the recordings. Thus, we have a coherent collection of songs that reflects Dutch everyday song culture in the early 20th century. Virtually all of these songs have a stanzaic structure. Each stanza repeats the melody, and each stanza

[1] Available from: http://www.liederenbank.nl/mtc.

consists of a number of phrases. Both in the transcriptions and in the song books, phrase endings are indicated. Figure 1 shows a typical song from the collection. The language of the songs is standard Dutch with occasionally some dialect words or nonsense syllables. All songs were digitally encoded by hand at the Meertens Institute (Amsterdam) and are available in Humdrum **kern format. The phrase endings were encoded as well and are available for computational analysis and modeling.

3. OUR APPROACH

Our general approach is to isolate trigrams from the melodies and to label those as either cadential or non-cadential. A cadential trigram is the last trigram in a phrase. We compare two kinds of trigrams: trigrams of successive notes (*note-trigrams*), and trigrams of successive pitches (*pitch-trigrams*), considering repeated pitches as one event. In the case of pitch-trigrams, a cadence pattern always consists of the three last unique pitches of the phrase. There are two reasons for including pitch-trigrams. First, pitch repetition is often caused by the need to place the right number of syllables to the melody. It occurs that a quarter note in one stanza corresponds to two eighth notes in another stanza because there is an extra syllable at that spot in the song text. Second, in models of closure in melody [11, 15] successions of pitches are of primary importance.

Figure 1 depicts all pitch-trigrams in the presented melody. The trigram that ends on the final note of a phrase is a cadential trigram. These are indicated in bold. Some cadential trigrams cross a phrase boundary when the next phrase starts with the same pitch.

From each trigram we extract a number of feature values that reflect both melodic and textual properties. We then perform a classification experiment using a Random Forest Classifier [2]. This approach can be regarded a 'bag-of-trigrams' approach, where each prediction is done independently of the others, i.e. all sequential information is lost. Therefore, as a next step we take the labels of the direct neighboring trigrams into account as well. The final classification is then based on a majority vote of the predicted labels of adjacent trigrams. These steps will be explained in detail in the next sections.

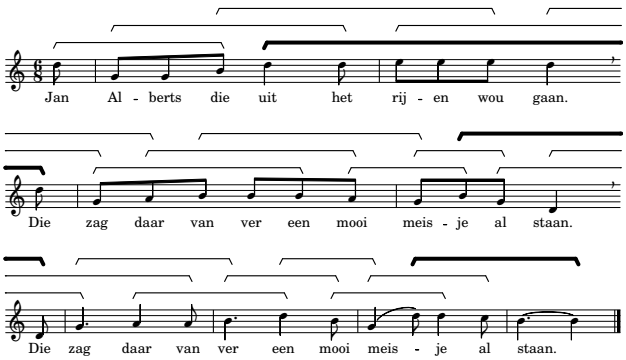

Figure 1. Examples of pitch-trigrams. The cadential trigrams are indicated in bold.

4. FEATURES

We represent each trigram as a vector of feature values. We measure several basic properties of the individual pitches and of the pattern as a whole. The code to automatically extract the feature values was written in Python, using the music21 toolbox [5]. The features are divided into groups that are related to distinct properties of the songs. Some features occur in more than one group. The following overview shows all features and in parentheses the value for the first trigram in Figure 1. Detailed explanations are provided in sections 4.1 and 4.2.

Pitch Features

Scale degree Scale degrees of the first, second, and third item (5, 1, 3).

Range Difference between highest and lowest pitch (4).

Has contrast third Whether there are both even and odd scale degrees in the trigram (False).

Contour Features

Contains leap Whether there is a leap in the trigram (True).

Is ascending Whether the first and second intervals, and both are ascending (False, True, False).

Is descending Whether the first and second intervals, and both are descending (True, False, False).

Large-small Whether the first interval is large and the second is small (True).

Registral change Whether there is a change in direction between the first and the second interval (True).

Rhythmic Features

Beat strength The metric weights of the first, second and third item (0.25, 1.0, 0.25).

Min beat strength The smallest metric weight (0.25).

Next is rest Whether a rest follows the first, second and third item (False, False, False).

Short-long Whether the second item is longer than the first, and the third is longer than the second (False, False).

Meter The meter at the beginning of the trigram ("6/8").

Textual Features

Rhymes Whether a rhyme word ends at the first, second and third item (False, False, False).

Word stress Whether a stressed syllable is at the first, second and third item (True, True, True).

Distance to last rhyme Number of notes between the last the first, second and third item and the last rhyme word or beginning of the melody (0, 1, 2).

Narmour Closure Features

Beat strength The metric weights of the first, second and third item (0.25, 1.0, 0.25).

Next is rest Whether a rest follows the first, second and third item (False, False, False).

Short-long Whether the second item is longer than the first, and the third is longer than the second (False, False).

Large-small Whether the first interval is large ($\geq$ fifth) and the second is small ($\leq$ third) (True).

Registral change Whether there is a change in direction between the first and the second interval (True).

Contextual Features

Next is rest third Whether a rest or end of melody follows the third item (False).

Distance to last rhyme Number of notes between the last the first, second and third item and the last rhyme word or beginning of the melody (0, 1, 2).

4.1 Melodic Features

Several of the features need some explanation. In this section we describe the melodic features, while in the next section, we explain how we extracted the textual features.

HasContrastThird is based on the theory of Jos Smits-Van Waesberghe [15], the core idea of which is that a melody gets its tension and interest by alternating between

pitches with even and uneven scale degrees, which are two contrasting series of thirds.

The metric weight in the Rhythmic features is the beat-strength as implemented in music21's meter model.

The Narmour features are based on the six (preliminary) conditions of closure that Narmour states at the beginning of his first book on the Implication-Realisation theory [11, p.11]: "[...] melodic closure on some level occurs when 1. a rest, an onset of another structure, or a repetition interrupts an implied pattern; 2. metric emphasis is strong; 3. consonance resolves dissonance; 4. duration moves cumulatively (short note to long note); 5. intervallic motion moves from large interval to small interval; 6. registral direction changes (up to down, down to up, lateral to up, lateral to down, up to lateral, or down to lateral). Of course, these six may appear in any combination." Because the melodies are monophonic, condition 3 has no counterpart in our feature set.

The contextual features are not features of the trigram in isolation, but are related to the position in the melody. In an initial experiment we found that the distance between the first note of the trigram and the last cadence is an important predictor for the next cadence. Since this is based on the ground-truth label, we cannot include it directly into our feature set. Since we expect rhyme in the text to have a strong relation with cadence in the melody, we include the distance to the last rhyme word in number of notes.

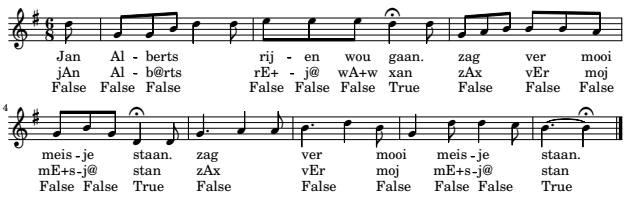

Figure 2. Rhyme as detected by our method. The first line shows the original text after removing non-content words. The second line shows the phonological representations of the words (in SAMPA notation). The third line shows whether rhyme is detected ('True' if a rhyme word *ends* at the corresponding note).

4.2 Textual Features

In many poetical texts, phrase boundaries are determined by sequences of rhyme. These establish a structure in a text, both for aesthetics pleasure and memory aid [14]. In folk music, phrasal boundaries established by sequences of rhyme are likely to relate to phrases in the melody.

We developed a rhyme detection system which allows us to extract these sequences of rhyming lyrics. Because of orthographical ambiguities (e.g. *cruise*, where /u:/ is represented by *ui* whereas in *muse* it is represented by *u*), it is not as straightforward to perform rhyme detection on orthographical representations of words. Therefore, we transform each word into its phonological representation (e.g. *cruise* becomes /kru:z/ and *bike* /baɪk/).

_	_	_	c	r	u	i	'k
_	_	c	r	u	i	s	r
_	c	r	u	i	s	e	u:
c	r	u	i	s	e	_	0
r	u	i	s	e	_	_	z
u	i	s	e	_	_	_	0

Figure 3. Example sliding window for phoneme classification.

We approach the problem of phonemicization as a supervised classification task, where we try to predict for each character in a given word its corresponding phoneme. We take a sliding window-based approach where for each focus character (i.e. the character for which we want to predict its phonemic representation) we extract as features n characters to the left of the focus character, n characters to the right, and the focus character itself. Figure 3 provides a graphical representation of the feature vectors extracted for the word *cruise*. The fourth column represents the focus character with a context of three characters before and three after the focus character. The last column represents the target phonemes which we would like to predict. Note that the first target phoneme in Figure 3 is preceded by an apostrophe ('k), which represents the stress position on the first (and only) syllable in cruise. This symbolic notation of stress in combination with phonology allows us to simultaneously extract a phonological representation of the input words as well as their stress patterns. For all words in the lyrics in the dataset we apply our sliding window approach with $n = 5$, which serves as input for the supervised classifier. In this paper we make use of a $k = 1$ Nearest Neighbor Classifier as implemented by [6] using default settings, which was trained on the data of the e-Lex database [2]. In the running text of our lyrics, 89.5% of the words has a direct hit in the instance base, and for the remaining words in many cases suitable nearest neighbors were found. Therefore, we consider the phonemicization sufficiently reliable.

We assume that only content words (nouns, adjectives, verbs and adverbials) are possible candidate rhyme words. This assumption follows linguistic knowledge as phrases typically do not end with function words such as determiners, prepositions, etcetera. Function words are part of a closed category in Dutch. We extract all function words from the lexical database e-Lex and mark for each word in each lyric whether it is a function word.

We implemented rhyme detection according to the rules for Dutch rhyme as stated in [19]. The algorithm is straightforward. We compare the phoneme-representations of two words backwards, starting at the last phoneme, until we reach the first vowel, excluding schwas. If all phonemes

[2] http://tst-centrale.org/en/producten/lexica/e-lex/7-25

Class	pr	rec	F_1	σ_{F_1}	support
note-trigrams					
cadence	0.84	0.72	0.78	0.01	23,925
nocadence	0.96	0.98	0.97	0.01	183,780
pitch-trigrams					
cadence	0.85	0.69	0.76	0.01	23,838
nocadence	0.95	0.98	0.96	0.00	130,992

Table 1. Results for single labels.

Class	pr	rec	F_1	σ_{F_1}	support
note-trigrams					
cadence	0.89	0.72	0.80	0.01	23,925
nocadence	0.96	0.99	0.98	0.00	183,780
pitch-trigrams					
cadence	0.89	0.71	0.79	0.01	23,838
nocadence	0.95	0.98	0.97	0.01	130,992

Table 2. Results for classification with label trigrams.

and the vowel are exactly the same, the two words rhyme.

As an example we take *kinderen* ('children') and *hinderen* ('to hinder'). The phoneme representations as produced by our method are /kɪndərə/ and /hɪndərə/. The first vowel starting from the back of the word, excluding the schwas (/ə/), is /ɪ/. Starting from this vowel, the phoneme representations of both words are identical (/ɪndərə/). Therefore these words rhyme.

We also consider literal repetition of a word as 'rhyme', but not if a sequence of words is repeated literally, such as in the example in Figure 1. Such repetition of entire phrases occurs in many songs. Labeling all words as rhyme words would weaken the relation with cadence or 'end-of-sentence'. We only label the last word of repeated phrases as a rhyme word. Figure 2 shows an example.

5. CLASSIFICATION WITH SINGLE LABELS

As a first approach we consider the trigrams independently. A melody is represented as 'bag-of-trigrams'. Each trigram has a ground-truth label that is either 'cadence' or 'no cadence', as depicted in Figure 1 for pitch-trigrams'

We employ a Random Forest classifier [2] as implemented in the Python library scikit-learn [13]. This classifier combines n decision trees (*predictors*) that are trained on random samples extracted from the data (with replacement). The final classification is a majority vote of the predictions of the individual trees. This procedure has proven to perform more robustly than a single decision tree and is less prone to over-fitting the data. Given the relatively large size of our data set, we set the number of predictors to 50 instead of the default 10. For the other parameters, we keep the default values.

The evaluation is performed by 10-fold cross-validation. One non-trivial aspect of our procedure is that we construct the folds at the level of the songs, rather than at that of individual trigrams. Since it is quite common for folk songs to have phrases that are literally repeated, folding at the level of trigrams could result in identical trigrams in the train and test subsets, which could lead to an overfitted classifier. By ensuring that all trigrams from a song are either in the test or in the train subset, we expect better generalization. This procedure is applied throughout this paper.

The results are shown in Table 1. For both classes averages of the values for the precision, the recall and the F_1-measure over the folds are included, as well as the standard deviation of the F_1 measure, which indicates the variation over the folds. The number of items in both classes (*support*) shows that cadences are clearly a minority class.

We observe that the note-trigrams lead to slightly better cadence-detection as compared to pitch-trigrams. Apparently, the repetition of pitches does not harm the discriminability. Furthermore, there is an unbalance between the precision and the recall of the cadence-trigrams. The precision is rather high, while the recall is moderate.

6. CLASSIFICATION WITH LABEL TRIGRAMS

When our cadence detection system predicts the class of a new trigram, it is oblivious of the decisions made for earlier predictions. One particularly negative effect of this near-sightedness is that the classifier frequently predicts two (or even more) cadences in a row, which, given our our training material, is extremely unlikely. We attempt to circumvent this 'defect' using a method, developed by [20] that predicts trigrams of class labels instead of single, binary labels. Figure 4 depicts the standard single class classification setting, where each trigram is predicted independent of all other predictions. In the label trigram setting (see Figure 5), the original class labels are replaced with the class label of the previous trigram, the class label of the current trigram and the label of the next trigram. The learning problem is transformed into a sequential learning problem with two stages. In the first stage we predict for each trigram a label trigram $y^{(t)} = (y_1, y_2, y_3)$ where $y \in \{0, 1\}$. To arrive at the final single class predictions (i.e. is it a cadence or not), in the second stage we take the majority vote over the predictions of the focus trigram and those of its immediate left and right neighboring trigrams. Take t_4 in Figure 5 as an example. It predicts that the current trigram is a cadence. The next trigram and the previous trigram also predict it to be a cadence and based on this majority vote, the final prediction is that t_4 is a cadence. Should t_3 and t_5 both have predicted the zero class (e.g. $y^{(t_3)} = (0, 0, 0)$ and $y^{(t_5)} = (0, 1, 0)$), the majority vote would be 0. The advantage of this method is that given the negligible number of neighboring cadences in our training data, we can virtually rule out the possibility to erroneously predict two or more cadences in a row.

Table 2 shows the performance of the label-trigram classifier for both classes and both for pitch and note trigrams. The values show an important improvement for the precision of cadence-detection and a slight improvement of the recall. The lower number of false positives is what we expected by observing the classification of adjacent trigrams as 'cadence' in the case of the single-label classifier.

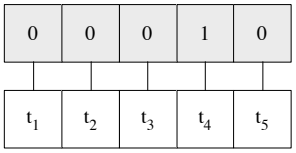

Figure 4. Short example sequence of trigrams. Each trigram t_i has a binary label indicating whether the trigram is cadential (1) or non-cadential (0).

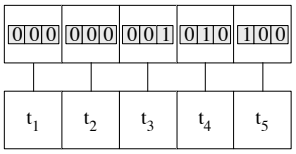

Figure 5. Label-trigrams for the same sequence as in Figure 1, where t_4 has label 1 and the other trigrams have label 0. Each trigram t_i gets a compound label consisting of its own label and the labels of the direct neighboring trigrams.

7. ABLATION STUDY

To study the importance of the various kinds of features, we perform an ablation study. We successively remove each of the groups of features as defined in section 4 from the full set and do a classification experiment with the remaining features. Subsequently, we perform a similar series of classification experiments, but now with each single group of features. The first series shows the importance of the individual groups of features, and the second series shows the predictive power for each of the groups. Because the groups are assembled according to distinct properties of music and text, this will give insight in the importance of various musical and textual parameters for cadence detection. We use the label-trigram classifier with the note-trigrams, which performed best on the full set.

We expect occurrence of rests to be a very strong predictor, because according to our definition a 'rest' always follows after the final cadence, and we know that in our corpus rests almost exclusively occur between phrases. Therefore, we also take the three features that indicate whether a rest occurs in the trigram or directly after it, as a separate group. The performance when leaving these three features out will show whether they are crucial for cadence detection.

Table 3 shows the evaluation measures for each of the feature subsets. Precision, recall and F_1 for class 'cadence' are reported. Again, the values are averaged over 10 folds.

We see that none of the single groups of features is crucial for the performance that was achieved with the complete set of features. The basic melodic features (F_{pitch}, $F_{contour}$, and $F_{rhyhmic}$) all perform very bad on their own, showing low to extremely low recall values. The contour features even do not contribute at all. Only the rhythmic features yield some performance. The features on rest are included in the set of rhythmic features. The classification with just the features on rest, F_{rest} shows very high precision and low recall. Still, the recall with all rhythmic features is higher than only using the rest-features. Since rests are so tightly related to cadences in our corpus, the high precision for F_{rest} is what we expected. If we exclude the rest-features, the precision stays at the same level as for the entire feature set and the recall drops with 0.06, which shows that only a minority of the cadences exclusively relies on rest-features to be detected.

Subset	pr	rec	F_1	σ_{F_1}
F_{all}	0.89	0.72	0.80	0.01
$F_{all} \setminus F_{pitch}$	0.88	0.72	0.79	0.01
F_{pitch}	0.84	0.04	0.08	0.01
$F_{all} \setminus F_{contour}$	0.88	0.73	0.80	0.01
$F_{contour}$	0.00	0.00	0.00	0.00
$F_{all} \setminus F_{rhythmic}$	0.79	0.49	0.60	0.01
$F_{rhythmic}$	0.90	0.35	0.50	0.01
$F_{all} \setminus F_{textual}$	0.85	0.58	0.69	0.02
$F_{textual}$	0.70	0.40	0.51	0.01
$F_{all} \setminus F_{narmour}$	0.83	0.55	0.66	0.01
$F_{narmour}$	0.95	0.30	0.45	0.01
$F_{all} \setminus F_{contextual}$	0.87	0.67	0.76	0.01
$F_{contextual}$	0.71	0.45	0.56	0.01
$F_{all} \setminus F_{rest}$	0.87	0.67	0.76	0.01
F_{rest}	0.97	0.27	0.43	0.02

Table 3. Results for various feature subsets for class 'cadence'.

The set of features that is based on the conditions of closure as formulated by Narmour shows high precision and low recall. Especially the high precision is interesting, because this confirms Narmour's conditions of closure. Apparently, most patterns that are classified as cadence based on this subset of features, are cadences indeed. Still, the low recall indicates that there are many cadences that are left undetected. One cause could be that the set of conditions as stated by Narmour is not complete, another cause could be the discrepancy between our features and Narmour's conditions. Further investigation would be necessary to shed light on this. Removing the Narmour-based features from the full feature set does not have a big impact. The other features have enough predictive power.

The textual features on their own show moderate precision and very moderate recall. They are able to discern certain kinds of cadences to a certain extent, while missing most of the other cadences. The drop of 0.14 in recall for $F_{all} \setminus F_{textual}$ as compared to the full set shows that text features are crucial for a considerable number of cadences to be detected. The same applies to a somewhat lesser extent to contextual features. Removing the contextual features from the full set causes a drop of 0.05 in the recall, which is considerable but not extreme. It appears that the group of cadence trigrams for which the contextual features are crucial is not very big.

8. CONCLUSION AND FUTURE WORK

In this paper we developed a system to detect cadences in Western folk songs. The system makes use of a Random Forest Classifier that on the basis of a number of handcrafted features (both musical and textual) is able to accurately locate cadences in running melodies. In a follow-up experiment we employ a method, originally developed for textual sequences, that predicts label-trigrams instead of the binary labels 'cadence' or 'non-cadence'. We show that incorporating the predictions of neighboring instances into the final prediction, has a strong positive effect on precision without a loss in recall.

In the ablation study we found that all groups of features, except for the contour features, contribute to the overall classification, while none of the groups is crucial for the majority of the cadences to be detected. This indicates that cadence detection is a multi-dimensional problem for which various properties of melody and text are necessary.

The current results give rise to various follow-up studies. A deeper study to the kinds of errors of our system will lead to improved features and increased knowledge about cadences. Those that were detected exclusively by textual features form a particular interesting case, possibly giving rise to new melodic features. Next, n-grams other than trigrams as well as skip-grams [7] could be used, we will compare the performance of our method with existing symbolic segmentation algorithms, and we want to make use of other features of the text such as correspondence between syntactic units in the text and melodic units in the melody.

9. REFERENCES

[1] Rens Bod. Probabilistic grammers for music. In *Proceedings of BNAIC 2001*, 2001.

[2] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[3] Bertrand H Bronson. Some observations about melodic variation in british-american folk tunes. *Journal of the American Musicological Society*, 3:120–134, 1950.

[4] Emilios Cambouropoulos. The local boundary detection model (lbdm) and its application in the study of expressive timing. In *Proc. of the Intl. Computer Music Conf*, 2001.

[5] Michael Scott Cuthbert and Christopher Ariza. Music21: A toolkit for computer-aided musicology and symbolic music data. In *Proceedings of the 11th International Conference on Music Information Retrieval (ISMIR 2010)*, pages 637–642, 2010.

[6] Walter Daelemans, Jakub Zavrel, Ko Van der Sloot, and Antal Van den Bosch. TiMBL: Tilburg Memory Based Learner, version 6.3, Reference Guide, 2010.

[7] David Guthrie, Ben Allison, W. Liu, Louise Guthrie, and Yorick Wilks. A closer look at skip-gram modelling. In *Proceedings of the Fifth international Conference on Language Resources and Evaluation LREC-2006*, 2006.

[8] David Huron. *Sweet Anticipation*. MIT Press, Cambridge, Mass., 2006.

[9] Zoltán Juhász. Segmentation of hungarian folk songs using an entropy-based learning system. *Journal of New Music Research*, 33(1):5–15, 2004.

[10] Ilmari Krohn. Welche ist die beste Methode, um Volks- und volksmässige Lieder nach ihrer melodischen (nicht textlichen) Beschaffenheit lexikalisch zu ordnen? *Sammelbände der internationalen Musikgesellschaft*, 4(4):643–60, 1903.

[11] Eugene Narmour. *The Analysis and Cognition of Basic Melodic Structures - The Implication-Realization Model*. The University of Chicago Press, Chicago and Londen, 1990.

[12] Marcus Pearce, Daniel Müllensiefen, and Geraint Wiggins. The role of expectation and probabilistic learning in auditory boundary perception: A model comparison. *Perception*, 39(10):1365–1389, 2010.

[13] F. Pedregosa et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[14] David C. Rubin. *Memory in Oral Traditions*. Oxford University Press, New York, 1995.

[15] Jos Smits van Waesberghe. *A Textbook of Melody: A course in functional melodic analysis*. American Institute of Musicology, 1955.

[16] B. Suchoff. *Preface*, pages ix–lv. State University of New York Press, Albany, 1981.

[17] W. Suppan and W. Stief, editors. *Melodietypen des Deutschen Volksgesanges*. Hans Schneider, Tutzing, 1976.

[18] David Temperley. A probabilistic model of melody perception. In *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR 2006)*, Victoria, BC, 2006.

[19] Erica van Boven and Gillis Dorleijn. *Literair Mechaniek*. Coutinho, Bussum, 2003.

[20] Antal Van den Bosch and Walter Daelemans. Improving sequence segmentation learning by predicting trigrams. In *Proceedings of the Ninth Conference on Natural Language Learning, CoNLL-2005*, pages 80–87, Ann Arbor, MI, 2005.

[21] Frans Wiering and Hermi J.M. Tabachneck-Schijf. Cognition-based segmentation for music information retrieval systems. *Journal of New Music Research*, 38(2):137–154, 2009.