Natural language processing of lyrics

Conference Paper · January 2005

DOI: 10.1145/1101149.1101255 · Source: DBLP

5 authors, including:

Pedro Cano

50 PUBLICATIONS **1,270** CITATIONS

SEE PROFILE

Fabien Gouyon

Institute for Systems and Computer Engineer…

95 PUBLICATIONS **1,582** CITATIONS

SEE PROFILE

NATURAL LANGUAGE PROCESSING of LYRICS

Jose P. G. Mahedero
Music Technology Group-IUA
Universitat Pompeu Fabra
Ocata 3, 08003 Barcelona, Spain
jpgarcia@iua.upf.es

Álvaro Martínez
Music Technology Group-IUA
Universitat Pompeu Fabra
Ocata 3, 08003 Barcelona, Spain
amartinez@iua.upf.es

Pedro Cano
Music Technology Group-IUA
Universitat Pompeu Fabra
Ocata 3, 08003 Barcelona, Spain
pcano@iua.upf.es

ABSTRACT

We report experiments on the use of standard natural language processing (NLP) tools for the analysis of music lyrics. A significant amount of music audio has lyrics. Lyrics encode an important part of the semantics of a song, therefore their analysis complements that of acoustic and cultural metadata and is fundamental for the development of complete music information retrieval systems. Moreover, a textual analysis of a song can generate ground truth data that can be used to validate results from purely acoustic methods. Preliminary results on language identification, structure extraction, categorization and similarity searches suggests that a lot of profit can be gained from the analysis of lyrics.

Categories and Subject Descriptors

H.5.5 [**Information Interfaces And Presentation**]: Sound and Music Computing

General Terms

Algorithms

Keywords

Music information retrieval, Automatic music classification, Lyrics processing

1. INTRODUCTION

There is a great interest in accessing music contents nowadays. Besides the search by editorial data such as artist or song name, users can navigate in human derived genre taxonomies or follow recommendations derived from collaborative filtering systems. Collaborative filtering exploit information of the type: "People who listened to X tend to like Y". Additionally, technologies that analyze music content are being pursued. For example, a number of algorithms that describe music from audio are developed [2, 7, 5].

MM'05, November 6–11, 2005, Singapore.

We report on preliminary experiments on the use of basic natural language processing tools for the analysis of music lyrics. A great deal of popular songs have lyrics, which are relatively easily to obtain from a number of on-line sites. Part of the semantic of an audio musical piece resides exclusively in its lyrics [8]. Their analysis hence complement what can be learned from collaborative, cultural or acoustic analysis [2]. As pointed by [7] a significant amount of music searches refers directly to the lyrics: around 28.9% of searches query using lyrics' fragments and 2.6% addresses to song's storyline [7]. Also, a relevant part of music searches relate to information that can be relatively easily inferred from lyrics, such as nationality 12.5% or affect (whether the song is funny, silly, plaintive) 2.4% [7].

Off-the-shelf natural language processing (NLP) algorithms [9], such as those used in similarity between texts, can be powerful lyrics navigation tools as well as a hint for artist or even plagiarism identification. Logan et al. [6] and Baumann [2] have previously exploited lyrics information for music similarity browsing. In this paper we extend their experiments to other NLP tasks, namely language identification, structure detection or text categorization.

Another motivation for our experiment is the use of NLP algorithms as a tool for ground truth metadata creation. NLP techniques can sometimes be more accurate then their audio counterparts (think for example of the problem of identifying the language of the lyrics). Accordingly, information extracted from lyrics automatically can be used as a source of metadata useful for training and validating audio algorithms. Another example of superior performance of text-based methods is that of structure detection. Manual audio segmentation is very time consuming and the accuracy of the task using purely audio methods is lower than that of lyrics automatic segmentation.

2. EXPERIMENTS

It is relatively simple to find lyrics on the WWW. There are many free websites where a simple crawler can download this material, e.g.: `http://www.lyrics.com` or `http://www.lyrics4u.com` to name a few. We made experiments in four different areas: language recognition, structure extraction, thematic categorization and similarity searches.

Table 1: bi-tri gram frequency examples

bi-tri gram	Spanish	English
ixt	0.26530	0.15436
biz	0.00043	0.00361
buj	0.00175	0.01851
ita	0.34588	0.03630
zo	0.658682	0.25000
sod	0.00089	0.00194

Table 2: Language identification

Type of identification	results	accuracy
song title	136/180	75%
song lyrics	460/500	92%

2.1 Language identification

2.1.1 Algorithm

We have used the Perl module *Lingua::Ident*. It is available at `http://www.cpan.org`. The algorithm is based on Ted Dunning' statistical identification algorithm [3].
For a good quality in recognition, not only the size but also the quality of the training text play a role. On its training phase, the algorithm uses the technique of *bi/tri-grams*. Bi/Trigrams are groups of two/three letters which are calculated from the training text. We chose a excerpt from an European parliament session transcription translated to 8 languages (Danish, Dutch, English, French, German, Italian, Portuguese and Swedish) in order to have good corpora and stored bi and trigrams with their correspondent frequency into files. See Table 1. Those files are initially loaded in memory, then the input text is divided into bi and tri-grams which are compared to the initial ones. We obtained a probability for each possible language, then we return the maximum a posteriori probability.

2.1.2 Lyrics language identification

For the evaluation of the language identification module, we selected 500 lyrics from 5 different languages: Spanish, Italian, French, German and English. 92% of identification were correct, see Table 2.

2.1.3 Title language identification

We were very interested in investigating the performance of the language identification when using short input text such as the title of a song. This small size of the text is a problem for language identifying and misleads the language identification module so much that only 75% (see Table 1) of the results were correct. However, this preliminary result suggests that this simple module can be very useful combined with pure acoustic similarity methods. Even though it does make more sense to be applied to lyrics, lyrics are not always available. It is much easier to, at least, have the title, the album or the name of the artist.

2.2 Structure extraction

There exist a correlation between music and lyric structure. As well as music, lyrics are divided in to the same parts:

Introduction (Intro): usually one verse composed by three or four phrases used to introduce the main theme or to give a context to the listener

Verse: verse roughly corresponds with a poetic stanza. Lyrics in verses tend to repeat less than they do in choruses.

Chorus: the refrain of a song. It assumes a higher level of dynamics and activity. When two or more sections of a lyric have almost identical text, these sections are instances of the chorus. A verse repeats at least twice with none or little differences between repetitions, becoming then, the most repetitive part of a lyric. It is also where the main theme is more explicit. As well as what happens with music, it is also the part which listeners tend to remember.

Bridge: In song writing, a bridge is an interlude that connects two parts of that song. As verses repeat at least twice, the bridge may then replace the 3rd verse or follow it thus delaying the chorus. In both cases it leads into the chorus.

Outro: not always present, this part is located at the end

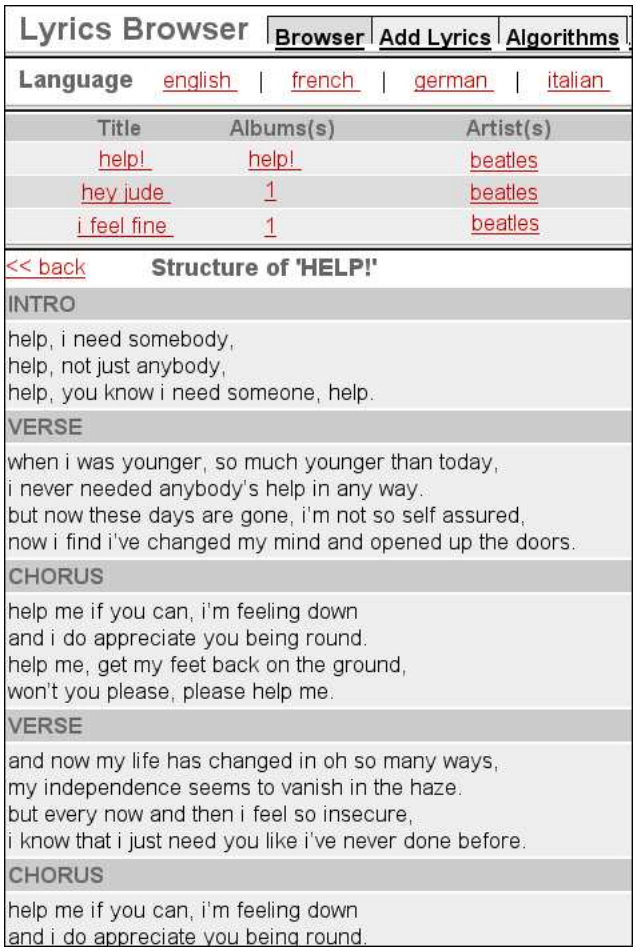

Figure 1: Sample web application for showing the results of the experiments. Structure extraction for Beatles' *Help!*

of a lyric and tends to be a conclusion about the main theme.

The algorithm used initially works with lyrics having a clearly recognizable structure (which is not always the case) divided into paragraphs. The strategy is based on weighting all of the paragraphs following results given by *descriptors* used and then tag them with a label describing it.

The list of descriptors include:

- Full length text
- Paragraphs in which lyric is explicitly divided
- Absolute and relative position of each paragraph in the lyric
- Number of lines or verses of each paragraph
- Paragraph similarity (measured with the cosine distance)
- Percentages over the whole lyric e.g.: percentages of *chorus* versus percentages of *verses*

Our algorithm has three steps: Descriptor extraction, Temporal tags hypothesis and then final tagging:

1. Descriptor extraction: text is normalized and divided into paragraphs. For each of them, descriptors are computed and Perl module String::Approx (`http://www.CPAN.org`) is used to compute similarity between paragraphs. Depending on results of similarity, the problem is classified in three different types:

   - type 0: paragraphs are very similar between them
   - type 1: paragraphs are completely different
   - type 2: some are similar and some not

   Obviously, type 2 is the most usually found. Step one is repeated several times by relaxing the similarity threshold until similar paragraphs are obtained.

2. Temporal tags hypothesis: this step is in charge of proposing a temporal solution to the problem. It tries to detect the most easily-identifiable parts of the lyric such as main chorus (they are usually the most repeated parts) or some verses having no similar one.

3. Final tagging: in this step every unit of the lyric is disambiguated. It is based on a set of standard compositional rules, e.g.: it is most probable for a song to start with an introduction or a verse rather than start with a chorus. It adds a score to each part used in case of needing to disambiguate.

We tested the segmentation algorithm against 30 lyrics, 6 for each language, which had previously been manually segmented. The algorithm yielded 76.66% of accuracy (units correctly segmented and identified). the results of each language are show in Table 3. See fig 1 for an example of structure segmentation for Beatles' *Help!* on a sample application for showing the results of the experiments.

Table 3: Structure Extraction

Language	results	accuracy
English	5/6	83%
French	4/6	66%
German	5/6	83%
Italian	5/6	83%
Spanish	4/6	66%
Overall	23/30	76%

2.3 Thematic categorization

In this experiment, the goal was to build a classifier able to classify lyrics into 5 distinct categories, namely: Love, Violent, Protest (antiwar), Christian and Drugs. Algorithms able to identify violent or explicit lyrics are obviously useful to maybe filter its access to children. As Christian music examples exist in all the major popular music styles (from pop to heavy metal), it is a good example of kind of songs that can benefit from such an algorithm, the most obvious way of identifying Christian music is by the lyrics.

The thematic categorizer that we used is a classical probabilistic classifier method known as Naive Bayes [4]. It classifies a new instance of a document D from a finite set C of predetermined classes. Given a set of words $W(D) = \{w_1, w_2 \dots w_n\}$ it computes the probability of D to belong to category C as follows $P(C|W) = \frac{P(W|C)P(C)}{P(W)}$, where $P(C)$ is the prior probability of category C and $P(W|C)$ is the conditional probability for word W given category C. Based on data observed on each experiment, the probability of a set of words given a category and the probability of the category can be computed. With that information, the category which maximizes the value for the following expression is selected.

$$Best = argmax_c \frac{P(W|C)\,P(C)}{P(W)} \tag{1}$$

The algorithm was implemented using Perl module *AI::Categorizer*. The corpus for this experiment consisted of 125 songs manually divided into the above mentioned 5 categories. The Naive Bayes classifier yielded a 82% on a 10-fold crossvalidation.

2.4 Similarity searches

Searching for similar lyrics is an interesting way of navigating on music collections. To obtain a similarity measure, we used the standard cosine distance that starts by computing a vector for each document as follows: $v(D) = [w_1, w_2 \dots w_n]$ where the w_j come from the classical measure Inverse Document Frequency:

$$w_{ij} = f_{ij} \log\left(\frac{N}{n_j}\right) \tag{2}$$

where N is the total number of documents, n_j is the number of documents containing term j and f_{ij} is the frequency of term j in the ith document.

The cosine distance d_{nq} between vectors n and q d_{nq} is :

$$d_{nq} = t_n.t_q = \frac{\sum w_{nj} w_{qj}}{||t_n||||t_q||} \tag{3}$$

Now we can define a vector for each document by concatenating distance between document q and all documents

on the corpus:

$$\Delta_q = [d_{1q} d_{2q} \dots d_{nq}] \quad (4)$$

With this measure we built an algorithm to compare documents and made two different types of tests in order to find similarities between songs and similarities between an artist and his/her songs.

First of all, a test was made between versions of the same song. Eagles' "Hotel California") is a song with very well defined lyrics (Eagles' only made one version of it) and we got a relevance of 98% with 4 versions of different bands.

A second experiment was made with a very versioned song such as "Sweet Jane" from the band "The velvet underground". In this case, the authors themselves recorded different versions, live one, extended one, etc. with small changes on lyrics. We obtained a relevance of 82% for 4 versions either from "The Velvet Underground" or other bands.

Finally we tried to find similarities between Queen's song "I want to break free" and a database of 5,000 lyrics. The results we got were that the most similar one had a normalized (0 to 1) relevance of 0.62, which can be explained by the fact that lots of songs include words like "break" of "free".

3. DISCUSSION

3.1 Language identification

This is from far the most accurate feature on lyrics processing provided there is enough input text to be recognized. In cases were lyrics are written in more than one language (for example Beatles' Michelle) or with onomatopoeic or non sense words (for example Santana's Jin-Go-Lo-Ba), recognition tends to fail. Language identification using shorter pieces of text is much less accurate and results in errors, specially among languages with the same roots. Identification errors are more frequent between Latin languages such as Spanish, Portuguese or Italian from one side, and English and German on the other side. There are even many words that are written the same, so titles with only one of these words may, inevitably, confuse the identifier module. Despite this cultural handicap, results are surprisingly good with 75% accuracy.

3.2 Structure extraction

Structure extraction is one of the most subjective aspects, even if it is not fully reliable yet, it is suitable for bootstrapping audio segmentation algorithms.

3.3 Thematic categorization

Definition of the initial training categories for thematic categorization is very subjective, but result determining task. Even if sometimes words or phrases are fit best in one category (for example "God" or "Jesus"') there are more which are context dependent. For example the phrase "you are my love" fits in almost every category, and sometimes it's hard to disambiguate and assign a category for a lyric.

3.4 Similarity searches

The margin among cover versions and plagiarism is narrower for lyrics than from audio. This is the main fact for obtaining not so good results in similarity between lyrics of different songs. It works very well between versions of the same song even if there is not an official version. This can be a very useful technology for right management entities which registers audio and lyrics.

4. CONCLUSIONS

We reported experiments in four different areas: language identification, thematic categorization, structure extraction and similarity searches. Although the results showed some of the techniques like structure extraction or thematic categorization can be still improved, its performance, together with the ubiquity of lyrics and relative ease with which they can be grabbed from on-line repositories, suggests that Natural Language Processing techniques are going to be increasingly deployed. Lyrics indeed can be a good complement to acoustic and cultural metadata. In the future we would like to test the overlap of lyrics similarity with the other types of similarity. We believe that Natural Processing Language techniques can be successfully used for the creation of extensive ground truth metadata for the evaluation of pure audio content-based methods.

5. ACKNOWLEDGMENTS

This work is partially funded by the European Union to the SIMAC IST-FP6-507142 project (`http://www.semantic-audio.org`)

6. ADDITIONAL AUTHORS

Markus Koppenberger and Fabien Gouyon
Music Technology Group-IUA
Universitat Pompeu Fabra
Ocata 3, 08003 Barcelona, Spain
koppi@iua.upf.es and fgouyon@iua.upf.es

7. REFERENCES

[1] A. Berenzweig, B. Logan, D. Ellis and B. Whitman, A large-scale evaluation of acoustic and subjective music similarity measures, Proc. of the ISMIR,2003

[2] S. Baumann, Cultural Metadata for Artist Recommendation, Proc. of the WEDELMUSIC, 2003

[3] T. Dunning Statistical Identification of Language, Proc. of the WEDELMUSIC, ACM Trans. Program. Lang. Syst, 15, 5, 745-770, 1993

[4] Machine learning in automated text categorization, ACM Computing surveys, 1, 1, 1-47, 2002

[5] B. Logan, D. Ellis and A. Berenzweig, Towards Evaluation Techniques for Music Similarity, IEEE ICME, 2003

[6] B. Logan, A. Kositsky and P. Moreno, Semantic analysis of song lyrics, IEEE ICME, 2004

[7] D. Bainbridge, S.J. Cunningham and J.S. Downie, Analysis of queries to a Wizard-of-Oz MIR system: Challenging assumptions about what people really want, IEEE ICME, 2003

[8] Besson, M. and Faïta, F. and Peretz, I. and Bonnel, A.-M. and Requin, J., Singing in the brain: Independence of Lyrics and Tunes, Psychological Science, 494-498, 6, 9, 1998

[9] Manning, C. and Schütze, H., Foundations of statistical nature language processing, MIT Press, 1999