Acta Univ. Sapientiae, Informatica **10,** 2 (2018) 158–182

DOI: 10.2478/ausi-2018-0009

Connecting the Last.fm Dataset to LyricWiki and MusicBrainz. Lyrics-based experiments in genre classification

Zalán BODÓ
Babeş–Bolyai University, Faculty of Mathematics and Computer Science
Cluj-Napoca, Romania
email: zbodo@cs.ubbcluj.ro

Eszter SZILÁGYI
Cluj-Napoca, Romania
email: bordieszter@gmail.com

Abstract. Music information retrieval has lately become an important field of information retrieval, because by profound analysis of music pieces important information can be collected: genre labels, mood prediction, artist identification, just to name a few. The lack of large-scale music datasets containing audio features and metadata has lead to the construction and publication of the Million Song Dataset (MSD) and its satellite datasets. Nonetheless, mainly because of licensing limitations, no freely available lyrics datasets have been published for research.

In this paper we describe the construction of an English lyrics dataset based on the Last.fm Dataset, connected to LyricWiki's database and MusicBrainz's encyclopedia. To avoid copyright issues, only the URLs to the lyrics are stored in the database. In order to demonstrate the eligibility of the compiled dataset, in the second part of the paper we present genre classification experiments with lyrics-based features, including bag-of-n-grams, as well as higher-level features such as rhyme-based and statistical text features. We obtained results similar to the experimental outcomes presented in other works, showing that more sophisticated textual features can improve genre classification performance, and indicating the superiority of the binary weighting scheme compared to tf–idf.

Computing Classification System 1998: H.3.3, H.5.5, I.2.6, I.2.7
Mathematics Subject Classification 2010: 68T05, 68T50, 68U15
Key words and phrases: music information retrieval, lyrics dataset, music genre classification

1 Introduction

A central problem of music information retrieval (MIR) is the similarity search of music tracks. Since in the last two decades online music streaming services and music stores have become exceedingly popular, to facilitate the search for similar music, recommender systems became vitally important too. Automatic genre classification is considered an equally important problem, since classification arises in simply browsing music by genre information as well as in music recommendation.

The lack of large-scale music datasets containing audio features and metadata has lead to the construction and publication of the Million Song Dataset[1] (MSD) [3] and its satellite datasets. However, as pointed out in [37] or [17], no freely available large-scale lyrics dataset has yet been published for research, mainly due to copyright problems. Although the musiXmatch dataset[2] offers hundreds of thousands of music tracks with lyrics given as bag-of-words vectors [50], this representation narrows down its applicability.

In this paper we describe the compilation of an English lyrics dataset based on the Last.fm Dataset, connected to LyricWiki's database and MusicBrainz's encyclopedia. Because of the copyright issues mentioned earlier, the dataset does not explicitly contain song lyrics, but LyricWiki page URLs pointing to the lyrics. Beside the URL we also included the MusicBrainz ID of the track, if found, together with album and release year information. In order to demonstrate the eligibility of the compiled dataset we conducted genre classification experiments with lyrics-based features, including bag-of-n-grams as well as higher-level features such as rhyme-based and statistical text features. We obtained results similar to the experimental outcomes presented in other works, showing that sophisticated textual features can improve genre classification performance, and indicating the superiority of the binary weighting scheme compared to tf–idf (term frequency $\times$ inverse document frequency).

The remainder of this paper is structured as follows. In Section 2—without striving for completeness—we review the works related to our research: MIR datasets, lyrics collections and classification experiments performed using these sets. Section 3 describes the process underlying the construction of the dataset: the databases involved in the compilation procedure, the scheme of the dataset, as well as some statistics. In Section 4 we present genre classification experiments based on the lyrics of the music tracks, using bag-of-words, n-grams, rhyme-based and statistical text features. Section 5 presents the con-

[1] http://labrosa.ee.columbia.edu/millionsong/

[2] http://labrosa.ee.columbia.edu/millionsong/musixmatch

crete experimental settings and results, while Section 6 discusses the results and concludes the paper.

2 Related work

Although the need for large-scale music information databases is of increasing concern, only a few such resources are accessible for research or commercial applications. One of the largest collection made available for MIR is the Million Song Dataset [3] and its numerous complementary datasets. The work [20] surveys the state-of-the-art problems in music analysis, and thus, it is a thorough collection of related bibliographical references and datasets. Another recent and comprehensive work on MIR is [25], likewise containing a large bibliography and references to associated datasets.

The authors of [4] have demonstrated by EEG experiments that the lyrics and tunes of a song are processed independently in the brain, therefore one can deduce that using textual features from the lyrics may improve the performance of a genre classification system. Song lyrics evidently contain valuable information—even the absence of the lyrics is an important clue when guessing music genre. All the information we obtain from the lyrics as textual data are inherently present in the audio signal. However, extracting lyrics directly from the audio data is still a very difficult task [15]. Therefore, we rely on the different versions that can be found in specific databases or on the Internet, the results of independent voluntary transcription procedures undertaken by different persons, in most cases. Thus, it is not uncommon to find a few differences because of different spellings, marking of chorus or verses, annotation of background voices, abbreviations, censored words, etc. In [26] the alignment of song lyrics is accomplished by multiple sequence alignment in order to eliminate typographical errors. These and related problems, however, can be overcome by community maintenance [48]. Hence, using a community-maintained lyrics database such as LyricWiki might prove to be more accurate.

It is important to mention the seminal work of [59] on genre recognition based on audio features. The paper introduces the famous GTZAN dataset[3], which despite of its inaccuracies [55] it is widely accepted and used. Other prevalent collections are the ISMIR 2004[4] and the CAL500[5] datasets. We also mention here some of the recent works on audio feature based music genre

[3] http://marsyasweb.appspot.com/download/data_sets/
[4] http://ismir2004.ismir.net/genre_contest
[5] http://labrosa.ee.columbia.edu/millionsong/pages/additional-datasets

recognition using convolutional neural networks [30, 12, 14, 47, 8], deep neural networks [53], deep belief networks [21] and multiscale approaches [13]. For an excellent presentation of these and similar approaches we direct the reader to [24].

In [38] the problems of music genre classifications are studied and analyzed: ambiguities and subjectivity inherent to genre, album rather than individual recording labeling, relatively frequent emergence of new genres, etc. It is also emphasized the importance of assigning multiple genre labels to music tracks, as this would result in a more realistic evaluation of classification systems. Bag-of-words features are combined with rhyme, part-of-speech (POS) and statistical text features in [36] for genre classification. The experiments are performed on a collection of 397 randomly sampled songs distributed among ten genres. The source of the lyrics data was not revealed in the paper. In [31] genre classification in the MSD is performed based on various feature types: audio (timbre, loudness and tempo), textual (bag-of-words, emotional valence) and combined features. Learning is accomplished via regularized multiclass logistic regression. The work [17] presents genre and best vs. worst music classification and release date prediction experiments using n-gram features extended with other higher-level features, including POS tags, rhymes, echoisms, semantic fields, etc. The experiments are carried out on a dataset built by the authors specifically for the targeted classification tasks, in which lyrics, genre information, album ratings and release dates were obtained from different online databases. The F_1 scores obtained in our experiments are very similar to their results.

The differences between poetry, song lyrics and other articles are studied in [54] using the adjectives extracted from the text. The presented method is also able to differentiate between poetic lyricists and non-poetic ones. The source for the lyrics data is not specified in the article. The authors of [10] perform lyrics-based mood prediction in the MSD using various term weighting schemes and find no statistically significant differences in the accuracy results. In [9] music subject classification based on lyrics and user interpretations are compared. The data was obtained from songmeanings.com and songfacts.com. Mood classification is studied in [33] using the lyrics of music tracks. The authors also study the relation between features and emotions to identify the most discriminative features for each quadrants. The lyrics data used in the experiments was collected using lyrics.com, ChartLyrics and MaxiLyrics, the tracks being annotated manually. The work [56] discusses evaluation approaches in music genre recognition, but also contains a useful list of existing datasets.

As the number of recent publications show, the lyrics collection provided by LyricFind[6]—through a signed research agreement—is becoming more and more popular. This is usually used together with the iTunes Search API[7] to obtain genre and other meta-information about the songs. It also has a bag-of-words version similar to musiXmatch, containing the bag-of-words representation of 275 905 lyrics.[8] In [16] lexical novelty of song lyrics is studied, and the authors find the already suspected fact that top-100 music is less lexically innovative than less popular music. A lyrics-based network is built to analyze musical relationships over time in [2]. It is observed that self-reference correlates highly with influence, the most central genres being jazz, pop and rock. LyricFind's collection is used in [58] as well, and a hierarchical attention network is applied to classify genre based on song lyrics in two scenarios, using 117 and 20 classes, respectively. The learning model allows to inspect the importance of words, lines and segments in lyrics.

The recent work [60] presents the construction of the ALF-200k dataset including 176 audio and lyrics features of more than 200 000 music tracks, together with their occurrence statistics in user playlists. Using the different sets of features the authors perform playlist membership prediction by adding random tracks originally not belonging to the playlist. Connecting with other databases it would be intriguing to perform genre classification experiments using these features too.

As related work shows, since no standard lyrics dataset can be found to work with, almost every study uses its own data, comparison between different methods being utterly complicated. This was the main reason behind building the collection connecting the Last.fm Dataset to LyricWiki and MusicBrainz. LyricWiki was chosen over other similar databases because of the advantages of community maintenance. To avoid copyright issues, instead of the actual verses only the LyricWiki URLs of the lyrics were included. We also publish unigram, bigram and trigram versions of this dataset, i.e. containing the n-gram representation of the lyrics. In order to validate the usage of the compiled dataset, genre classification experiments are presented in the second part of the present paper.

[6] http://lyricfind.com/
[7] http://apple.co/1qHOryr
[8] https://www.smcnus.org/lyrics/

3 Construction of the lyrics dataset

3.1 The Million Song and the Last.fm Dataset

The Million Song Dataset is a free collection of audio features and metadata for one million contemporary music tracks. It was released for research purposes in 2011 by the Laboratory for the Recognition and Organization of Speech and Audio (LabROSA) department of the Columbia University[9] in collaboration with The Echo Nest[10]. MSD is more than a single dataset, it is also a cluster of many spin-off datasets[11]: SecondHandSongs (cover songs), musiXmatch (lyrics), Last.fm (song-level tags and similarity), Taste Profile (user data), thisismyjam-to-MSD mapping (user data), tagtraum genre annotations (genre labels), Top MAGD dataset (genre labels).

The Last.fm Dataset[12] is a complementary set of MSD, containing song tags and similarity information, built in collaboration with Last.fm[13], an online music database and music recommendation system. Last.fm also provides an API for metadata retrieval[14], also used by us to connect and extend the Last.fm Dataset.

Last.fm data (i.e. tags, musical samples, etc.) has been used in numerous experiments. An interesting work we mention is [35], in which the authors analyze the evolution of popular music and musical revolutions identifiable in the collected data, using data mining techniques such as latent Dirichlet allocation and novelty detection.

3.2 LyricWiki

LyricWiki[15] is a community-maintained lyrics database, offering music metadata services, released in 2006.

In March 2013 it was the seventh largest MediaWiki installation[16], and as of August 2018 contains over two million pages. LyricWiki also provided a web API for searching songs and lyrics, however, due to licensing restrictions, in

[9] http://labrosa.ee.columbia.edu/
[10] http://the.echonest.com/
[11] http://labrosa.ee.columbia.edu/millionsong/pages/additional-datasets
[12] http://labrosa.ee.columbia.edu/millionsong/lastfm
[13] http://www.last.fm
[14] http://www.last.fm/api
[15] http://lyrics.wikia.com/wiki/LyricWik
[16] https://en.wikipedia.org/wiki/LyricWiki

2016 the API has been discontinued.[17] Interestingly, as of December 2018, the API[18] is again functional.

3.3 MusicBrainz

MusicBrainz[19] is an open online music encyclopedia of music metadata launched in 2000 [57]. As of 2018, the database, more precisely its *recording* index, contains over 19 million entries, being one of the largest such databases. MusicBrainz provides a web service[20] for metadata retrieval too.

The web API was used by us to obtain additional release information about a song.

3.4 Building the dataset

The Last.fm Dataset contains 839 122 training and 104 212 test records.[21] We succeeded in using 224 762 (199 217 training and 25 545 test) data, i.e. we managed to find the lyrics of that many songs in LyricWiki's database.

The dataset consists of tracks, where every track is identified by a unique Echo Nest ID. Beside the ID, every track has the following fields: *artist*, *title*, *timestamp*, *similars*, *tags*. The timestamp stores the date of creation. Similar tracks are enumerated as a list of tuples, containing the ID of the proximal track along with a similarity, a scalar value between 0 and 1. Similarly, the assigned tags are given as a list of tuples, consisting of a tag name, e.g. "rock", and a relevance value, an integer between 0 and 100.[22]

The *timestamp* and *similars* fields were removed from the tracks, however, if needed, one can easily retrieve this information by connecting our dataset to the Last.fm Dataset using the track ID. Only tags having a relevance value greater than or equal to 50 have been kept. Such a step is motivated by the fact that tracks have been tagged quite freely by the Last.fm users, therefore, one can also find some strange ones, as shown in Table 1. Thus, we considered a tag relevant only if at least 50% of the time it was assigned to the track.

[17] In 2015, using the API, we managed to connect the Last.fm Dataset to LyricWiki pages, using the artist's name and the title of the song.

[18] http://lyrics.wikia.com/api.php

[19] http://musicbrainz.org/

[20] http://musicbrainz.org/doc/Development/XML_Web_Service/Version_2

[21] The dataset had been downloaded on 12.05.2016 and had contained a total of 943 334 files, 13 tracks less than the value published on the official site of the dataset.

[22] A value of r means that in r% of the cases the respective tag was assigned to the track by the users.

what I want to hear at my funeral	super happy feel good
vagany	one of the best solos
amaaazzzinnnggg	songs to fall asleep to in a good way
banging the head on the wall	betterfriend
soooooo beautiful I died again	holy riffs

Table 1: Some random tags from the Last.fm Dataset with frequency 1.

Artist queried:	Queen & David Bowie
Song queried:	Under Pressure (Rah Mix) (Radio Edit) (1999 Digital Remaster)
Artist returned:	Queen
Song returned:	Under Pressure

Table 2: Answer returned by LyricWiki for track `TRTTPMY128F4258EAC`.

We encountered only one case where the relevance value did not exceed this threshold and it happened for track `TRQXIYJ128F930A292` from the training set—we left this record in the dataset with its maximum-valued tag.[23]

3.4.1 The lyrics

The *url* field contains the LyricWiki link of the lyrics. This LyricWiki page URL was obtained by using the LyricWiki API. LyricWiki returns the artist name, the song title, a short snippet of the lyrics and the link to the page containing the full lyrics and song information. There were three cases when the respective track was omitted from the dataset: not found, instrumental or not English. The language of the lyrics and the release information can be deduced from the page of the full lyrics.

Because lyrics of musical tracks are proprietary work, in most of the cases its publication are forbidden, therefore, we only offer the LyricWiki page where the lyrics can be extracted from, and n-gram datasets (up to trigrams) from which the lyrics cannot be reconstructed.

The artist name and song title returned by LyricWiki API can differ from the queried data, probably because of the preprocessing steps built into the search engine. Though we have not found any documentation regarding the

[23] The track in question is Bobby Brown's song, 'Pretty Little Girl', and is assigned only two tags, namely 'killer shredding' with a relevance score 2 and 'mod psych' along with a value of 0.

indexing/search operations, we have found evidence of text normalization.[24] An example is shown in Table 2. Because of these differences we decided to also store the returned data in the *artist_new* and *title_new* fields, respectively.

3.4.2 Release information and MusicBrainz IDs

We decided to extend the dataset by including release information (album and release year) for each song, and also to store the MusicBrainz IDs[25] (MBIDs) of the tracks.

The MSD comes with additional databases including a metadata SQLite database[26], containing metadata information such as song title, release, year, etc. Our *album* and *year* fields correspond to the *release* and *year* fields of this database. Sometimes release information occurred on LyricWiki pages, suggesting also that the respective track appeared on multiple releases. If found, using the first release mentioned on the page, this forms the content of the *album_new* field. In case this information was not to be found on the LyricWiki page, we made additional efforts to obtain it from the MusicBrainz encyclopedia. In order to connect MusicBrainz, we first performed a search with the Last.fm API using the artist name and song title from the Last.fm dataset, and then another search using the retrieved song and artist information by LyricWiki. In this way we obtained two MusicBrainz identifiers for each track, and stored it in the *mbid* and *mbid_new* fields, respectively. Knowing the MBID of a music track it is simple to query its releases, from which we stored the title of the first one in the *album_new* field.

For getting the release year of the track we acted similarly: if the year was found on the LyricWiki page, that one was stored in the *year_new* field, otherwise it was queried from MusicBrainz.

In some cases differences in artist names, song and album titles are due to slight spelling discrepancies. However, of course, incomplete information—on either side—can also cause it. The differences may also arise from multiple releases of the same song: original song/original album, live edition/concert album, remixed version of the song, compilation album, etc. Errors, mismatches can also appear in such databases. The used databases, however, were not

[24] For example, searching for 'Déjà Vu' by 'The Tear Garden', 'Deja Vu' is found and returned—of which, surprisingly, the returned form without diacritical marks is the correct song title (https://www.discogs.com/Tear-Garden-Tired-Eyes-Slowly-Burning/master/7843).

[25] https://musicbrainz.org/doc/MusicBrainz_Identifier

[26] http://labrosa.ee.columbia.edu/millionsong/pages/getting-dataset

```
"TRFDMMO128F424D545": {
 "mbid":       "ed3dccc3-e47b-4c81-90be-fdd7e820647a",
 "mbid_new":   "ed3dccc3-e47b-4c81-90be-fdd7e820647a",
 "title":      "6:00",
 "title_new":  "6:00",
 "artist":     "Dream Theater",
 "artist_new": "Dream Theater",
 "album":      "Awake",
 "album_new":  "Awake",
 "year":       "1994",
 "year_new":   "1994",
 "url":        "http://lyrics.wikia.com/Dream_Theater:6:00",
 "tags":       [["Progressive metal", "100"]]
}
```


Figure 1: Sample data from our dataset.

checked against a ground-truth dataset, therefore, no such information can be reported by us.

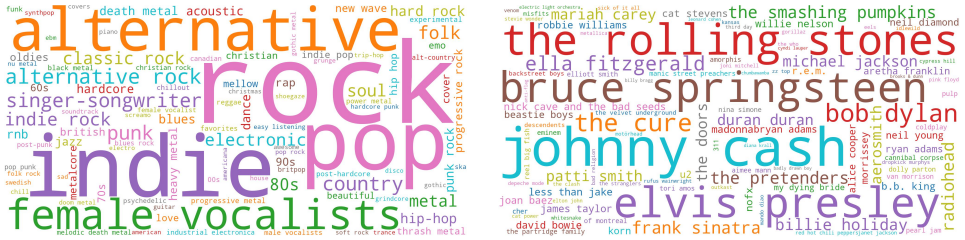
Summing it up, the Echo Nest ID being the key, the fields of a record in the dataset are the following:

- *mbid* – MusicBrainz ID returned by the Last.fm API for the artist and track name as given in the Last.fm Dataset (or MSD)
- *mbid_new* – MusicBrainz ID from the Last.fm API for the artist and track name as returned by LyricWiki
- *title* – title of the song according to MSD
- *title_new* – title of the song returned by LyricWiki
- *artist* – artist according to MSD
- *artist_new* – artist returned by LyricWiki
- *album* – album/release name according to MSD
- *album_new* – album/release name extracted from LyricWiki/using the Last.fm and MusicBrainz API
- *year* – release year according to MSD
- *year_new* – release year extracted from LyricWiki/using the Last.fm and MusicBrainz API
- *url* – LyricWiki URL of the song's lyrics
- *tags* – list of the tags assigned to the track filtered by the relevance value (≥ 50)

A sample data is shown in Figure 1.

	Training data	Test data
album	0	0
album_new	6858	1088
year	31 452	4323
year_new	34 363	4787
mbid	27 254	3446
mbid_new	14 070	2137

Table 3: Counts of missing fields in the dataset.



(a) (b)

Figure 2: Word clouds of the (a) 100 most frequent tags and (b) 100 most popular artists.

3.5 Removal of duplicates and some statistics

As mentioned at the beginning of Section 3.4, linking the Last.fm dataset with LyricWiki pages we obtained 224 762 track records. According to millionsong's blog entry *"The 921,810 song dataset – duplicates"*[27], duplicate songs have been found in the MSD, which also affects the Last.fm Dataset. Using the official duplicate list of the MSD[28], we removed 27 529 records from the training and 3164 records from the test dataset. Therefore, our final dataset contains 171 688 and 22 381 (a total of 194 069) records.[29]

Table 3 shows the statistics of missing fields in the dataset.

In Figure 2 the most frequent tags and artists are shown, while the histograms of Figure 3 show the distribution of lyrics over years—as expected, a

[27] http://labrosa.ee.columbia.edu/millionsong/blog/11-3-15-921810-song-dataset-duplicates

[28] http://labrosa.ee.columbia.edu/millionsong/sites/default/files/AdditionalFiles/msd_duplicates.txt

[29] The dataset, under research-only, non-commercial license, is available for download at http://www.cs.ubbcluj.ro/~zbodo/lastfm.html.

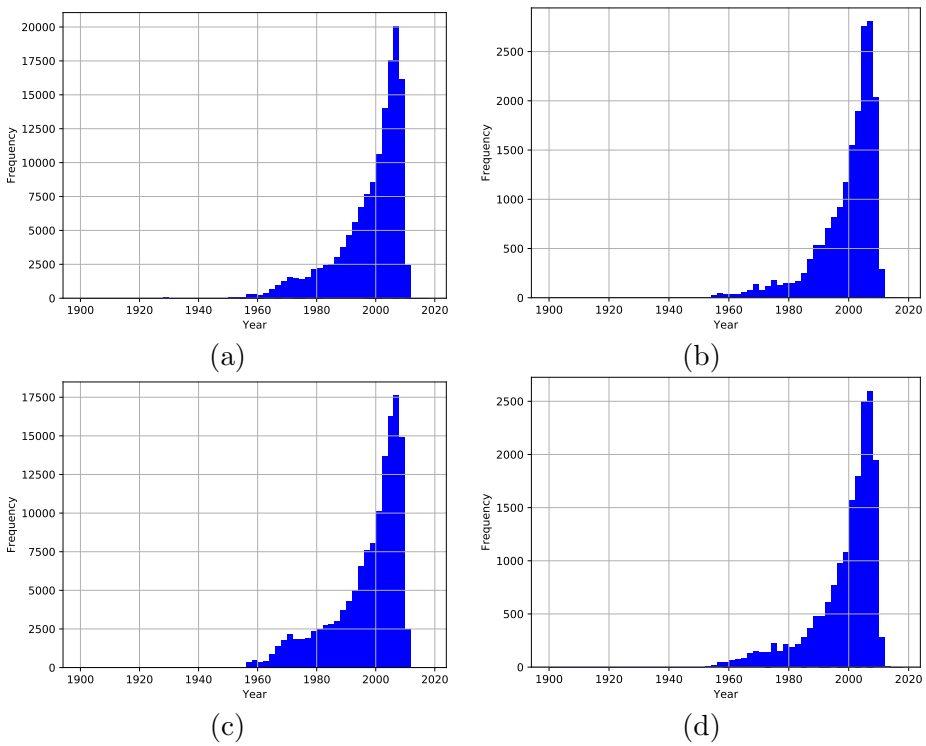


Figure 3: Distribution of lyrics over years in: (a) training data using *year*, (b) test data using *year*, (c) training data using *year_new*, (d) test data using *year_new*.

steep increase can be observed in the amount of available lyrics over time.

4 Lyrics-based genre classification

Determining the genre of a music track is considered to be an important task in MIR, which can be viewed as a special case of the music similarity problem. To assign genre labels to a song is a difficult task even for human annotators, because there are no clear and precise definitions of genres, thus often yielding subjective classifications. However, we mention that genre information is usually assigned to an artist or an album, rather than to a musical piece, which would be preferable. As pointed out in [38], musical genre classification should be based on numerous, complex features, including low-level, e.g. timbre-based features, but also high-level, e.g. cultural features. In [4] the au-

thors conducted experiments to demonstrate whether the lyrics and tunes of a song are processed independently in the brain. The analyzed electroencephalogram recordings showed that semantic and musical incongruities indeed do not affect each other. Hence, assuming that the lyrics can contain genre-related information, using lyrics-based textual features may have beneficial effects on classification.

For a general and also detailed discussion of music genre classification see [38].

In this section we describe a lyrics-based genre classification approach similar to [36, 31, 17]. The main goal of this experiment is to demonstrate the utility of the compiled dataset.

4.1 Choosing the genres

In order to use the dataset in some experiments the problem of musical genre classification, or more generally thematic categorization of lyrics was chosen [32, 38, 36, 17]. As it was already shown in Tables 1 and 2(a), the dataset contains a large variety of tags, ranging from genre related to other diverse descriptive labels. More precisely, the 194 069 records are assigned a number of 76 746 different tags, using the reduced tag lists. To perform a supervised learning task, we decided to choose only a subset of these, possibly denoting musical genres.

Determining a good taxonomy of musical genres is itself a difficult problem, and almost all online music stores and retailers use a different genre hierarchy. Thus, we were not able to find a suitable taxonomy consisting of a smaller number of meta genres, and decided to randomly select some of the most popular tags from our dataset (see Figure 2(a)). On how to derive better, objective genre taxonomies see the works [43] and [51].

Because of the almost limitless freedom given for the users in tagging, not all tags convey genre information about a song in the Last.fm Dataset. But, as described in Section 3.4, we used a threshold of 50 when deciding whether to keep a tag for a track. Thus, we expect that most of the time the remaining tags are valid, consisting of genre annotations and not deliberately misleading labels as described in [6].

The chosen tags—and the corresponding data counts—are the following:[30]

- *rap*: 2972 training, 451 test data (28th most popular tag)

[30] In order for the experiments to be reproducible, we mention that for each tag we required an exact match with case insensitivity.

- *reggae*: 1608 training, 178 test data (56th most popular tag)
- *jazz*: 3300 training, 339 test data (26th most popular tag)
- *punk*: 6760 training, 551 test data (8th most popular tag)
- *country*: 6360 training, 866 test data (9th most popular tag)
- *folk*: 6074 training, 831 test data (11th most popular tag)
- *pop*: 14267 training, 1855 test data (3rd most popular tag)
- *classic rock*: 6413 training, 482 test data (12th most popular tag)
- *electronic*: 5851 training, 851 test data (13th most popular tag)

The genres were chosen quite randomly, but some extra care was taken not to produce an extremely skewed distribution among the classes. This is the reason, for example, behind choosing *classic rock* instead of the *rock* label.

Thus, we are given a total of 60 009, i.e. 53 605 training and 6404 test data distributed unevenly among the 9 classes. The only non-overlapping class pairs are *reggae* and *country*, whilst the largest overlap of 643 (training and test) records happens between classes *pop* and *electronic*.

4.2 Choosing the features

4.2.1 N-grams

The bag-of-words model is a successful representation in information retrieval, which, despite its simplicity, yields surprisingly good results in categorizing text documents [50, 1]. The main drawback of the model is the assumption of independence between the words, but its effectiveness indicates that most of the time one can determine the category based on specific keywords, or more precisely by the distribution of these keywords. A somewhat better model that takes into account the word order is the n-gram representation [18]. In the experiments we successively extended the bag-of-words representation—i.e. the unigram model—with bigrams and trigrams.

4.2.2 Rhyme features

Can rhyme schemes be used to discriminate between musical genres? This is the question we wanted to answer by including rhyme features into the lyrics representation.

The Merriam-Webster dictionary gives us the definition of rhyme as the "correspondence in terminal sounds of units of composition or utterance (as two or

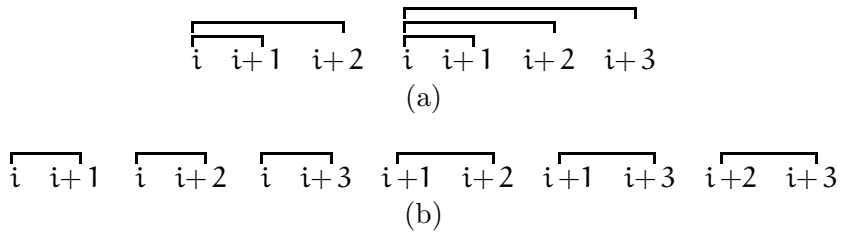


Figure 4: Rhyme features: (a) n-gram rhymes ($n_s = 3, n_e = 4$), (b) pairwise rhymes ($n_m = 4$).

more words or lines of verse)"[31], while the authors of [49] define rhyming as follows: "two words rhyme if their final stressed vowels and all following phonemes are identical". In our more primitive interpretation, two words rhyme if their last syllables are similar, and we will use this definition in building the rhyme features. The last two definitions, however, are incomplete: to find the rhyme scheme of a stanza one should also consider that a rhyme may span more words or syllables in line [22, 49]. Nonetheless, because of the relatively rare occurrences we decided not to consider these cases. Internal rhymes can also appear in song lyrics [22, 17], however, these do not influence the rhyme scheme. The authors of [17] used these kind of rhymes too to build lyrics-based features, calling them echoisms.

To find rhyming words in lyrics using our definition from above three components and a threshold are needed:

(a) hyphenator,
(b) phonetic algorithm,
(b) string similarity,

and a threshold for considering two syllables sufficiently similar. For determining the syllables the hyphenation method of OpenOffice and LibreOffice was used [42], representing the pronunciation of the last syllable was realized using the phonetic algorithm of Soundex [27], and for comparing these pronunciations we selected the Levenshtein distance [29, 19].

Two rhyme feature sets were used: an n-gram rhyme set, parametrized by n_s and n_e, and a pairwise rhyme set parametrized by n_m.[32] The n-gram rhyme set starts with rhyme schemes of length n_s and continues to build features

[31] Full definition of rhyme (2a), http://www.merriam-webster.com/dictionary/rhyme.

[32] The denomination *n-gram* is not the best here, since it does not fully reflect the nature of this feature, but hopefully the example given will clarify the vagueness.

	number of verses	number of rows	average no. of words in rows
rap	(9.5918,	72.7481,	8.7017)
classic rock	(7.1029,	33.0973,	6.6492)

Table 4: Example of statistical text feature vectors.

until length n_e, as shown in Figure 4(a). A feature is thus described by k binary values, $k = n_s - 1, n_s, \ldots, n_e - 1$, each value indicating the presence or absence of a rhyme. For example, from the alternating rhyme scheme ABAB—assuming it was correctly recognized—we obtain the following features with $n_s = 3$, $n_e = 4$: $(0, 1)$ (this will appear twice, because of ABA and BAB show the same pattern), $(0, 1, 0)$. The other rhyme feature set checks for pairwise rhymes between the i-th and $(i + k)$-th row of the lyrics independently, $k = 1, 2, \ldots, n_m - 1$, for all i. In contrast to the previous rhyme feature set, these features are encoded by pairs describing the distance between the row indices and the binary value showing whether or not a rhyme was found. Thus, using the same example as before, we get the following pairwise rhyme features using $n_m = 4$: $(1, 0)$ (three times, for indices $(0, 1)$, $(1, 2)$, $(2, 3)$), $(2, 1)$ (twice, for indices $(0, 2)$ and $(1, 3)$), and $(3, 0)$ (for the pair $(0, 3)$).

4.2.3 Statistical text features

Statistical text features—number of verses, average number of words in a row, etc.—are useful characteristics when classifying lyrics [36]. One could expect for example that the lyrics of a rap song is longer in average than the lyrics of a rock or pop song, which indeed turns out to be true. Comparing the averages of number of verses, number of rows and average number of words in rows between tracks of *rap* and *classic rock* we get the results shown in Table 4.

In our experiments we used the following 14 statistical text features: number of verses, number of rows, average number of words in rows, average word length, number of special characters (`!`, `.`, `?`, `:`, `;`, `-`, `,`, `'`, `"`), average frequency of numbers in the rows of the lyrics.

5 Experimental results

By performing the experiments we wanted to show the following: (i) n-gram features alone can yield good results, (ii) using rhyme and statistical text

Features	Micro F_1	Macro F_1
Unigrams, TF–IDF	48.47%	46.47%
Unigrams, frequency	50.02%	46.83%
Unigrams, binary	52.24%	50.30%
Unigrams (1000/class), binary	52.11%	49.44%
Uni + bigrams, TF–IDF	49.32%	48.03%
Uni + bigrams, frequency	50.26%	48.41%
Uni + bigrams, binary	54.50%	52.77%
Uni + bi + trigrams, TF–IDF	50.69%	48.56%
Uni + bi + trigrams, frequency	51.77%	49.42%
Uni + bi + trigrams, binary	**56.61**%	**54.53**%

Table 5: Micro and macro F_1 scores obtained using logistic regression.

Features	Micro F_1	Macro F_1
Rhyme + statistical text features	37.19%	22.44%
+ uni+bi+trigrams (binary)	**57.59**%	**54.99**%

Table 6: Micro and macro F_1 scores obtained using logistic regression with the new feature set alone and by augmenting the unigram, bigram and trigram feature set with it.

features can improve on the performance of the classifier. The goal was to approximately reproduce the experiments described in [36] and [17], and show the usefulness of the newly compiled dataset.

We applied a single-label classifier for learning [52], namely logistic regression [11, 5], using the *scikit-learn* Python library[33]. As mentioned in the previous section, we had slightly overlapping categories, which means that better results could have probably been achieved by using a ranking classifier and finding good thresholds for the categorization status values [52]. As for proving the above-mentioned two claims we needed no multilabel classifier. In our experiments, for every track the most frequent tag was used as its label. Thus, we are given 50 622 and 6113 test data.

To evaluate the models, micro- and macro-averaged F_1 scores were calculated [34].

Table 5 shows the results obtained using only n-gram features. We experimented with three weighting schemes: term frequency, tf–idf and binary weights [52, 34]. With unigrams we obtained 171 207 features, with unigrams

[33] http://scikit-learn.org/

	Genre	Uni+bi+trigram	Uni+bi+trigram+rhyme and stat. text features
1	rap	87.83%	88.36%
2	reggae	45.45%	42.67%
3	jazz	53.47%	54.07%
4	punk	48.33%	49.60%
5	country	65.26%	65.41%
6	folk	47.74%	47.61%
7	pop	61.79%	63.22%
8	classic rock	39.41%	40.63%
9	electronic	41.52%	43.34%
	Micro-averaged	56.61%	57.59%
	Macro-averaged	54.53%	54.99%

Table 7: F_1 scores for each genre for the unigram, bigram and trigram model and the same representation augmented with the rhyme and statistical text features.

and bigrams 3 845 117, while using unigrams, bigrams and trigrams together a total of 16 433 472 features were obtained.[34]

Table 6 shows the results obtained first by using the rhyme and statistical text features only, and in its second row the scores achieved by augmenting the n-gram document vectors with the rhyme and statistical text features. Table 7 lists the F_1 results for each genre separately.

The parameters of generating the features described in Section 4.2.2 were $n_s = 4$, $n_e = 5$, $n_m = 7$. Together with the statistical text features the new feature set has a cardinality of $36 + 14$. Finding the rhymes was performed using the Soundex algorithm.[35] The outputs of this algorithm, the phonetic representations of the input words—more precisely, of the last syllable of the input words—have to be used as inputs of a similarity or distance function with a predetermined threshold. For this we used normalized Levenshtein distance with threshold 0.7.[36] We mention that none of the parameters used in the experiments were selected using cross-validation or a similar procedure, therefore it is highly probable that by tuning the parameters better performance can be achieved.

[34] The unigrams training data has a sparsity of 5.68×10^{-4}%, the uni+bigrams 6.62×10^{-5}%, while using uni+bi+trigrams a sparsity of 2.59×10^{-5}% is observed.

[35] Fuzzy, https://pypi.python.org/pypi/Fuzzy.

[36] py_stringmatching, https://pypi.org/project/py_stringmatching/.

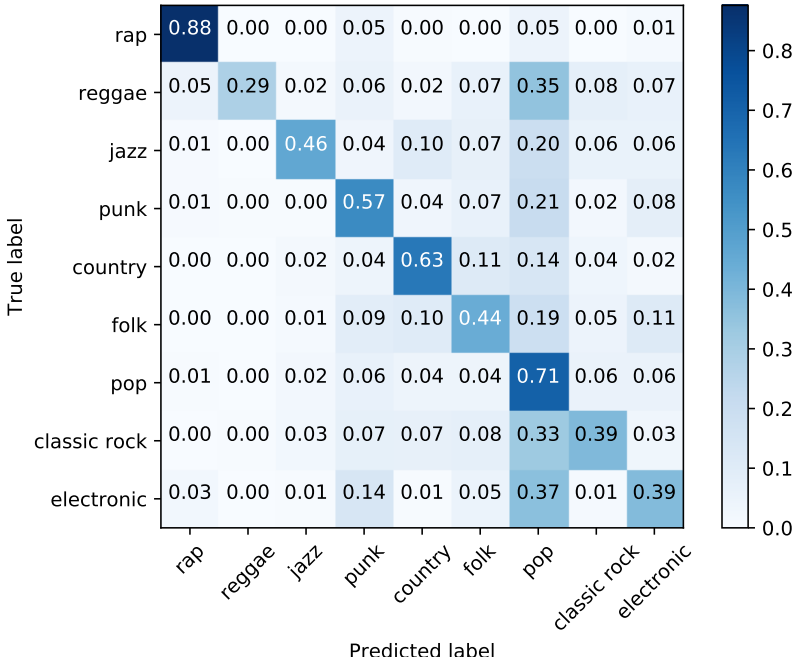
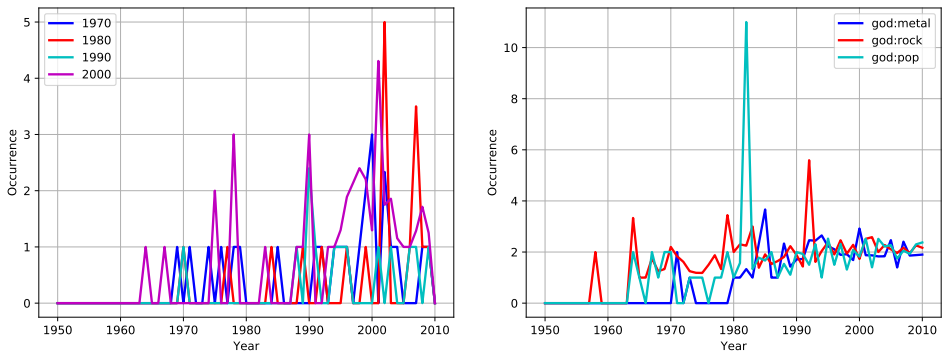


Figure 5: Confusion matrix for the uni+bi+trigram model augmented with rhyme and statistical text features. For the categories see Table 7.

6 Discussion and conclusions

We presented the compilation of a lyrics dataset linking the Last.fm Dataset, LyricWiki and MusicBrainz. The dataset contains the lyrics, i.e. LyricWiki URLs of English songs of the Last.fm Dataset (or MSD) found in LyricWiki's database, extended with additional release information. Knowing the MBID of a music track the dataset can be further extended with ease. After linking the Last.fm Dataset with LyricWiki and removing the duplicates—as described in Section 3.5—the final set contains 171 688 training and 22 381 test records. The tags of the music tracks were copied from the Last.fm Dataset without any text normalization, but the lists were thresholded at a relevance score ≥ 50. For the complete description of the database fields see Section 3.4.2.

In the second part of the paper we described the genre classification experiments conducted using the new dataset and considering some of the most frequent tags as genres. From previous research we already knew that using higher-level lyrics-based features improves on the performance of the genre



(a) (b)

Figure 6: (a) Occurrences of *1970*, *1980*, *1990* and *2000* and (b) the occurrence of *god* in lyrics from 1950 to 2010. The counts are normalized by the number of tracks per year.

prediction system, but we found a rather interesting fact regarding word features. Namely, that the best representation happens to be the binary weighting (see Table 5). These results suggest that the presence or absence of a term or n-gram is most likely a better indicator of the genre than the importance weighted distribution. This is similar to sentiment analysis, where binary word counts usually induce a better performance [23].

From the confusion matrix shown in Figure 5, we can see which genres are problematic to predict: reggae and pop, rock and pop, and electronic and pop are the most easily confusable in our system, while falsely predicting a track as being of pop genre is moderately high for every genre (see column 7 of the confusion matrix). One possible explanation of this phenomenon could be that indeed, analyzing the lyrics of these music genres, no significant differences can be found between them. Another explanation of the above is the proximity of the genres in question, for example in case of rock and pop. The Wikipedia article about rock music[37] says the following: "Like pop music, lyrics often stress romantic love but also address a wide variety of other themes that are frequently social or political." Also, in the article of pop music[38] we can find the following: "'Pop' and 'rock' were roughly synonymous terms until the late 1960s, when they became increasingly differentiated from each other." This might imply joining together some of the above categories and studying the labels of the misclassified tracks.

[37] https://en.wikipedia.org/wiki/Rock_music
[38] https://en.wikipedia.org/wiki/Pop_music

The constructed dataset can also be used in *culturomics* [39]. Figure 6 compares the occurrences of *1970*, *'80*, *'90* and *2000*, as well as the occurrence of *god* in lyrics from 1950 to 2010.

The compiled dataset was made publicly available to stay at the disposal of possible future MIR, psychological, linguistics, etc. research.[39] We publish the following datasets: (a) the dataset as described in Section 3.4.2 and (b) bag-of-n-grams representations of the lyrics, $n \in \{1, 2, 3\}$.

Though the primary goal of this paper was the description of the newly compiled LyricWiki-based dataset, lyrics-based genre classification can also be further studied in detail. A good starting point would be the more precise determination of rhymes, studying other phonetic algorithms like Metaphone and Double Metaphone [45, 46], or applying automatic rhyme detection methods as in [22]. Another direction would be the application of word and document embedding methods for lyrics representation [40, 41, 28, 44]. Since the parameters of our system were selected arbitrarily, a compulsory next step would be tuning these using cross-validation. Applying large-scale semi-supervised methods for learning [7] is also a possible future direction one can investigate. Finally, but not less important, we mention the assessment of the importance of different rhyme and statistical text features in predictions.

References

[1] C. Apté, F. Damerau, and S. M. Weiss. Toward language independent automated learning of text categorization models. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 23–30, Dublin, Ireland, 1994. Springer-Verlag. ⇒171

[2] J. Atherton and B. Kaneshiro. I said it first: Topological analysis of lyrical influence networks. In *ISMIR*, pages 654–660, 2016. ⇒162

[3] T. Bertin-Mahieux, D. P. W. Ellis, B. Whitman, and P. Lamere. The million song dataset. In A. Klapuri and C. Leider, editors, *ISMIR*, pages 591–596. University of Miami, 2011. ⇒159, 160

[4] M. Besson, F. Faita, I. Peretz, A.-M. Bonnel, and J. Requin. Singing in the brain: Independence of lyrics and tunes. *Psychological Science*, 9(6):494–498, 1998. ⇒160, 169

[5] C. M. Bishop. *Pattern recognition and machine learning*. Springer, 2006. ⇒174

[6] M. J. T. Carneiro. Towards the discovery of temporal patterns in music listening using Last.fm profiles. Master's thesis, Faculdade de Engenharia da Universidade do Porto, 2011. ⇒170

[39] See Section 3.5.

[7] O. Chapelle, B. Schölkopf, and A. Zien. *Semi-Supervised Learning*. The MIT Press, 2006. ⇒178

[8] K. Choi, Gy. Fazekas, M. Sandler, and K. Cho. Convolutional recurrent neural networks for music classification. In *ICASSP*, pages 2392–2396. IEEE, 2017. ⇒161

[9] K. Choi, J. H. Lee, X. Hu, and J. S. Downie. Music subject classification based on lyrics and user interpretations. In *Proceedings of the 79th ASIS&T Annual Meeting: Creating Knowledge, Enhancing Lives through Information & Technology*. American Society for Information Science, 2016. ⇒161

[10] H. Corona and M. P. O'Mahony. An exploration of mood classification in the million songs dataset. In *12th Sound and Music Computing Conference*, Ireland, 2015. Music Technology Research Group, Department of Computer Science, Maynooth University. ⇒161

[11] D. R. Cox. The regression analysis of binary sequences. *Journal of the Royal Statistical Society. Series B (Methodological)*, 2(2):215–242, 1958. ⇒174

[12] S. Dieleman, P. Brakel, and B. Schrauwen. Audio-based music classification with a pretrained convolutional network. In *ISMIR*, pages 669–674, 2011. ⇒161

[13] S. Dieleman and B. Schrauwen. Multiscale approaches to music audio feature learning. In *ISMIR*, pages 116–121, 2013. ⇒161

[14] S. Dieleman and B. Schrauwen. End-to-end learning for music audio. In *ICASSP*, pages 6964–6968. IEEE, 2014. ⇒161

[15] D. P. W. Ellis. Extracting information from music audio. *Communications of the ACM*, 49(8):32–37, 2006. ⇒160

[16] R. J. Ellis, Z. Xing, J. Fang, and Y. Wang. Quantifying lexical novelty in song lyrics. In *ISMIR*, pages 694–700, 2015. ⇒162

[17] M. Fell and C. Sporleder. Lyrics-based analysis and classification of music. In J. Hajic and J. Tsujii, editors, *COLING*, pages 620–631. ACL, 2014. ⇒159, 161, 170, 172, 174

[18] J. Fürnkranz. A study using n-gram features for text categorization, 1998. ⇒171

[19] W. H. Gomaa and A. A. Fahmy. A survey of text similarity approaches. *International Journal of Computer Applications*, 68(13):13–18, April 2013. ⇒172

[20] S. Gupta. Music data analysis: A state-of-the-art survey. *arXiv preprint arXiv:1411.5014*, 2014. ⇒160

[21] P. Hamel and D. Eck. Learning features from music audio with deep belief networks. In *ISMIR*, volume 10, pages 339–344, 2010. ⇒161

[22] H. Hirjee and D. G. Brown. Using automated rhyme detection to characterize rhyming style in rap music. *Empirical Musicology Review*, 5(4), 2010. ⇒172, 178

[23] D. Jurafsky and J. H. Martin. *Speech and language processing*. 2017. 3rd edition draft. ⇒177

[24] A. Kiss. Classification of hungarian folk music from Transylvania with convolutional neural networks. Master's thesis, Faculty of Mathematics and Computer Science, Babeş–Bolyai University, Romania, 2018. ⇒161

[25] P. Knees and M. Schedl. *Music Similarity and Retrieval*. Springer, Berlin–Heidelberg, 2016. ⇒160

[26] P. Knees, M. Schedl, and G. Widmer. Multiple lyrics alignment: Automatic retrieval of song lyrics. In *ISMIR*, pages 564–569, 2005. ⇒160

[27] D. E. Knuth. *The Art of Computer Programming, Vol. 3: Sorting and Searching*. Addison-Wesley, Reading, MA, 1973. ⇒172

[28] Q. Le and T. Mikolov. Distributed representations of sentences and documents. In *Proceedings of The 31st International Conference on Machine Learning*, pages 1188–1196, 2014. ⇒178

[29] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet physics doklady*, 10(8):707–710, 1966. ⇒172

[30] T. L. H. Li, A. B. Chan, and A. Chun. Automatic musical pattern feature extraction using convolutional neural network. In *Proc. Int. Conf. Data Mining and Applications*, 2010. ⇒161

[31] D. Liang, H. Gu, and B. O'Connor. Music genre classification with the million song dataset. Technical report, Machine Learning Department, CMU, 2011. ⇒161, 170

[32] J. P. G. Mahedero, A. Martinez, P. Cano, M. Koppenberger, and F. Gouyon. Natural language processing of lyrics. In *ACM Multimedia*, pages 475–478. ACM, 2005. ⇒170

[33] R. Malheiro, R. Panda, P. Gomes, and R. Paiva. Classification and regression of music lyrics: Emotionally-significant features. In *8th International Conference on Knowledge Discovery and Information Retrieval*, Porto, Portugal, 2016. ⇒161

[34] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*. Cambridge University Press, 2008. ⇒174

[35] M. Mauch, R. M. MacCallum, M. Levy, and A. M. Leroi. The evolution of popular music: USA 1960–2010. *Royal Society Open Science*, 2(5), 2015. ⇒163

[36] R. Mayer, R. Neumayer, and A. Rauber. Rhyme and style features for musical genre classification by song lyrics. In J. P. Bello, E. Chew, and D. Turnbull, editors, *ISMIR*, pages 337–342, 2008. ⇒161, 170, 173, 174

[37] R. Mayer and A. Rauber. Music genre classification by ensembles of audio and lyrics features. In A. Klapuri and C. Leider, editors, *ISMIR*, pages 675–680. University of Miami, 2011. ⇒159

[38] C. McKay and I. Fujinaga. Musical genre classification: Is it worth pursuing and how can it be improved? In *ISMIR*, pages 101–106, 2006. ⇒161, 169, 170

[39] J.-B. Michel, Y. K. Shen, A. P. Aiden, A. Veres, M. K. Gray, J. P. Pickett, D. Hoiberg, D. Clancy, P. Norvig, J. Orwant, S. Pinker, M. A. Nowak, and E. Lieberman Aiden. Quantitative analysis of culture using millions of digitized books. *Science*, 331:176–182, 2011. ⇒178

[40] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. ⇒178

[41] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013. ⇒178

[42] L. Németh. Automatic non-standard hyphenation in OpenOffice.org. *TUGboat*, 27(1):32–37, 2006. ⇒172

[43] F. Pachet and D. Cazaly. A taxonomy of musical genres. In J.-J. Mariani and D. Harman, editors, *RIAO*, pages 1238–1245. CID, 2000. ⇒170

[44] J. Pennington, R. Socher, and C. Manning. GloVe: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014. ⇒178

[45] L. Philips. Hanging on the metaphone. *Computer Language Magazine*, 7(12):38, December 1990. ⇒178

[46] L. Philips. The double metaphone search algorithm. *C/C++ Users Journal*, 18(6), June 2000. ⇒178

[47] J. Pons, T. Lidy, and X. Serra. Experimenting with musically motivated convolutional neural networks. In *CBMI*, pages 1–6. IEEE, 2016. ⇒161

[48] R. Priedhorsky, J. Chen, S. T. K. Lam, K. Panciera, L. Terveen, and J. Riedl. Creating, destroying, and restoring value in Wikipedia. In *Proceedings of the 2007 international ACM conference on Supporting group work*, pages 259–268. ACM, 2007. ⇒160

[49] S. Reddy and K. Knight. Unsupervised discovery of rhyme schemes. In *ACL (Short Papers)*, pages 77–82. The Association for Computer Linguistics, 2011. ⇒172

[50] G. Salton, A. Wong, and A. C. S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18:229–237, 1975. ⇒159, 171

[51] H. Schreiber. Improving genre annotations for the million song dataset. In M. Müller and F. Wiering, editors, *ISMIR*, pages 241–247, 2015. ⇒170

[52] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002. ⇒174

[53] S. Sigtia and S. Dixon. Improved music feature learning with deep neural networks. In *ICASSP*, pages 6959–6963. IEEE, 2014. ⇒161

[54] A. Singhi and D. G. Brown. Are poetry and lyrics all that different? In H.-M. Wang, Y.-H. Yang, and J. H. Lee, editors, *Proceedings of the 15th International Society for Music Information Retrieval Conference, ISMIR 2014, Taipei, Taiwan, October 27–31, 2014*, pages 471–476, 2014. ⇒161

[55] B. L. Sturm. An analysis of the gtzan music genre dataset. In *Proceedings of the second international ACM workshop on Music information retrieval with user-centered and multimodal strategies*, pages 7–12. ACM, 2012. ⇒160

[56] B. L. Sturm. A survey of evaluation in music genre recognition. In *International Workshop on Adaptive Multimedia Retrieval*, pages 29–66. Springer, 2012. ⇒161

[57] A. Swartz. MusicBrainz: a semantic Web service. *IEEE Intelligent Systems*, 17(1):76–77, 2002. ⇒164

[58] A. Tsaptsinos. Lyrics-based music genre classification using a hierarchical attention network. In *ISMIR*, pages 694–701, 2017. ⇒162

[59] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Transactions on speech and audio processing*, 10(5):293–302, 2002. ⇒160

[60] E. Zangerle, M. Tschuggnall, S. Wurzinger, and G. Specht. Alf-200k: Towards extensive multimodal analyses of music tracks and playlists. In *European Conference on Information Retrieval*, pages 584–590. Springer, 2018. ⇒162

Received: September 7, 2018 • Revised: December 5, 2018