EXPLOITING PLAYLISTS FOR REPRESENTATION OF SONGS AND WORDS FOR TEXT-BASED MUSIC RETRIEVAL

Chia-Hao Chung
National Taiwan University
b99505003@ntu.edu.tw

Yian Chen
KKBOX Inc.
annchen@kkbox.com

Homer Chen
National Taiwan University
homer@ntu.edu.tw

ABSTRACT

As a result of the growth of online music streaming services, a large number of playlists have been created by users and service providers. The title of each playlist provides useful information, such as the theme and listening context, of the songs in the playlist. In this paper, we investigate how to exploit the words extracted from playlist titles for text-based music retrieval. The main idea is to represent songs and words in a common latent space so that the music retrieval is converted to the problem of selecting songs that are the nearest neighbors of the query word in the latent space. Specifically, an unsupervised learning method is proposed to generate a latent representation of songs and words, where the learning objects are the co-occurring songs and words in playlist titles. Five metrics (precision, recall, coherence, diversity, and popularity) are considered for performance evaluation of the proposed method. Qualitative results demonstrate that our method is able to capture the semantic meaning of songs and words, owning to the proximity property of related songs and words in the latent space.

1. INTRODUCTION

Online music streaming services, such as Spotify, Apple Music, and KKBOX, create various playlists for the convenience of music listening for users. Meanwhile, users may create their own playlists for replay or music sharing with friends [1–5]. The use of playlist makes music retrieval and organization simple and easy, largely because the title of a playlist carries significant thematic information of the songs contained in the playlist [1–3]. The theme can be about an artist, genre, mood, or context of the playlist. Therefore, the thematic information is useful for music retrieval. The goal of this paper is to exploit playlists for text-based music retrieval.

One critical issue of text-based music retrieval is how to identify and quantify the relationship between words and songs (i.e. which songs and words are relevant to each other and how much is the relevance). Most previous approaches to text-based music retrieval rely on human-labeled datasets [6, 7] or social tags [8, 9] which normally have a limited size of vocabulary (word set). The web-based approach [10, 11] has been considered a good alternative because web documents have rich text information. However, its performance may degrade in the presence of noisy text [12]. In contrast, the playlist-based approach has the following appealing features: 1) The rich text information conveyed by the succinct playlist title is highly relevant to the songs in the playlist and 2) Songs wrapped in one playlist must be related to each other in a certain way. If the relationship can be determined from the playlist, additional efforts on audio signal analysis [6, 7, 11] can be saved.

Our main idea is to represent songs and words in a common latent space so that music retrieval can be converted to the problem of selecting songs sufficiently near the query word in the latent space. Specifically, we propose an unsupervised learning method to generate a representation of songs and words extracted from playlist titles, in which the learning function is optimized based on the co-occurrence of songs and words in playlists. As each song or word (an object) is represented as a vector in a latent space, the semantic similarity between two objects can be easily determined by the distance between the two corresponding vectors. By exploiting this property, we can improve the performance of text-based music retrieval.

Our contributions can be summarized as follows:

- We propose an unsupervised learning method to model the relevance between songs and words of playlists and to represent these two kinds of objects in a common latent space.
- We make text-based music retrieval easier to solve by formulating it as a nearest neighbor search problem in the latent space.
- Both qualitative and quantitative evaluations are conducted to demonstrate the effectiveness of the proposed method.

2. RELATED WORK

In this section, we review previous work related to playlist understanding, text-based music retrieval, and representation learning.

2.1 Playlist Understanding

To understand the use of playlist, Hagen [1] and Cunningham et al. [2] conducted user interviews to analyze various themes and contexts of playlists. The results motivated Pichl et al. [3] to mine common listening contexts using playlist titles for context-aware

© Chia-Hao Chung, Yian Chen, Homer Chen.
Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Chia-Hao Chung, Yian Chen, Homer Chen. “Exploiting Playlists for Representation of Songs and Words for Text-Based Music Retrieval”, 18th International Society for Music Information Retrieval Conference, Suzhou, China, 2017.

Playlist Title	Words	Songs (Artists)
Summer's Over	summer	The Boys of Summer (The Ataris) So Long, So Long (Dashboard Confessional) Last Days of Summer (Silverstein) Close To Home (The Get Up Kids) Always Summer (Yellowcard) …
Happy Morning Chill	happy morning chill	Snap Out Of It (Arctic Monkeys) Unbelievers (Vampire Weekend) Demons (Imagine Dragons) The Mother We Share (Chvrches) Everybody Wants To Rule The World (Lorde) …
George Michael - For the Heart	george_michael heart	Don't Let the Sun Go Down on Me (George Michael) Careless Whisper (George Michael) Heal The Pain (George Michael) A Different Corner (George Michael) I Can't Make You Love Me (George Michael) …

Table 1. Illustration of words extracted from playlists. Only the first five songs of a playlist are shown.

music recommendation. In our work, we take a step further and investigate how to exploit the words extracted from playlist titles for text-based music retrieval.

A related issue is playlist quality measurement. Motivated by the observation that the songs in a playlist, although diverse, are related to each other in a certain way, Fields [4] introduced coherence and diversity as metrics of playlist quality. It was found that popularity and freshness of songs in a playlist are also important metrics [5]. Considering that the response to a text query is in the form of playlist, we apply coherence, diversity, and popularity as metrics for performance evaluation.

2.2 Text-Based Music Retrieval

To allow the retrieval of music pieces by text query, the relevance between words and songs has to be identified. Turnbull et al. [6] and Chechik et al. [7] developed a multi-class classification approach to predict the relevance of a music piece to a query. To address the issue that the perception of relevance is subjective, Hariri et al. [8] and Cheng et al. [9] used a probabilistic model and listening records to personalize text-based music retrieval. To extend the coverage of text queries, Knees et al. [10–12] crawled web documents relevant to a music piece and represented the music piece by the text extracted from the web documents. However, most of the body of words contained in web documents can be irrelevant to the theme of the music pieces. To solve the problem, we develop an alternative approach that seeks relevant words from the playlist titles.

2.3 Representation Learning

Representation learning has been widely applied to music recommendation [13, 16, 29], playlist recommendation [17], music annotation and retrieval [18], playlist generation [19, 20], and listening behavior analysis [21, 22]. The popularity of representation learning is due to its two appealing features. First, it can efficiently handle large scale dataset [23, 24] because of low model complexity. Second, it makes information retrieval or recommendation an easy task that can be efficiently accomplished. However, little attention has been paid to exploit representation learning for text-based music retrieval. In this paper, we extend the idea of embedding learning [16–24], which is a typical representation learning approach, to model the relevance between songs and words of playlists.

3. PROPOSED METHOD

We first introduce the notations used in this paper. Then, we describe the proposed method for learning a representation of songs and words and the detail of the training processing, including optimization and data sampling. Finally, we describe how the learned representation is applied to text-based music retrieval.

3.1 Notations

Let $L = \{l_1, l_2, \dots, l_\mathrm{I}\}$ be a set of playlists and $T = \{t_1, t_2, \dots, t_\mathrm{I}\}$ be the set of corresponding playlist titles. Each playlist l_i, $1 \le i \le \mathrm{I}$, as illustrated in Table 1, is associated with a set of songs $S^i = \{s_1^i, s_2^i, \dots, s_{|l_i|}^i\}$ and a set of words $W^i = \{w_1^i, w_2^i, \dots, w_{|t_i|}^i\}$ extracted from t_i. Let $S = \{s_1, s_2, \dots, s_\mathrm{N}\}$ be the union of all S^i, and $W = \{w_1, w_2, \dots, w_\mathrm{M}\}$ be the union of all W^i. The goal is to learn a representation $\theta(\cdot)$ to map each $s_n \in S$ or $w_m \in W$ to a vector.

3.2 Representation Learning for Songs and Words

We extend the idea of embedding learning to songs and words. In its basic form, the embedding learning generates a representation for a set of objects based on the co-occurrence of the objects [23]. It consists of two stages. In the first (or initialization) stage, the representation $\upsilon(\cdot)$ assigns a vector of random values to each object. In second (or update) stage, the vector is progressively updated in two steps. In the first step, a conditional probability $P\big(o_c \big| \upsilon(o)\big)$ for each pair of objects o and o_c is created, where o_c is the co-occurring

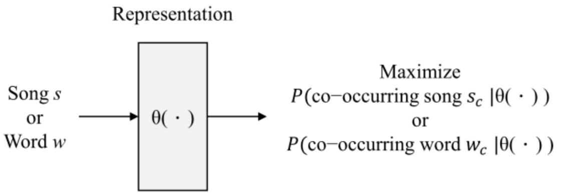

Representation
Song s
or
Word w
θ(·)
Maximize
P(co−occurring song s_c |θ(·))
or
P(co−occurring word w_c |θ(·))

Figure 1. Representation learning for songs and words. Given a song or a word, the representation is optimized based on its co-occurring song or word.

object of o. In the second step, $v(o)$ is optimized by maximizing the conditional probability. The two steps are repeated until the maximization for every pair of o and o_c is completed.

To extend the basic idea of embedding learning to songs and words, we need to define the co-occurring relationship between songs and words. If two songs (words) belong to the same playlist (playlist title), we say that they are co-occurring in the playlist (playlist title). Likewise, any word of a playlist title and any song in the playlist have a co-occurring relationship. In our method, co-occurring pairs of songs, words, or song and word are considered positive pairs.

Once the songs, words, and positive pairs of all playlists are in place, a learning process that consists of two stages is applied. In the first stage, the representation $\theta(\cdot)$ for each song or word is randomly initialized. In the second stage, $\theta(\cdot)$ is optimized. Specifically, we go through every positive pair and randomly select a word (or song), denoted as s (or w) from it. Then, we optimize the representation $\theta(s)$ (or $\theta(w)$) in two steps. In the first step, we construct a conditional probability which can be expressed in one of the following four formats:

$$P\big(s_c\big|\theta(s)\big), P\big(w_c\big|\theta(s)\big), P\big(w_c\big|\theta(w)\big), \text{or } P\big(s_c\big|\theta(w)\big),$$

where s_c or w_c is the remainder song or word in the positive pair. In the second step, we optimize the representation $\theta(s)$ (or $\theta(w)$) by maximizing the conditional probability. The two steps, as illustrated in Figure 1, are repeated until the maximization for every positive pair is completed (e.g. an epoch is completed).

We formulate the entire learning process by the following object function:

$$\mathcal{L} = \sum_{l_i \in L}\Big(\sum_{s \in S^i}\Big(\sum_{s_c \in S^i} \log P\big(s_c\big|\theta(s)\big) + \sum_{w_c \in W^i} \log P\big(w_c\big|\theta(s)\big)\Big) + \sum_{w \in W^i}\Big(\sum_{s_c \in S^i} \log P\big(s_c\big|\theta(w)\big) + \sum_{w_c \in W^i} \log P\big(w_c\big|\theta(w)\big)\Big)\Big). \quad (1)$$

Note that the natural logarithm converts a conditional probability to a log likelihood for the convenience of update stage [24]. The conditional probability $P\big(s_c\big|\theta(w)\big)$ is modeled by a softmax function [23] and can be rewritten as:

$$P\big(s_c\big|\theta(w)\big) = \frac{\exp(\varphi(s_c)\cdot\theta(w))}{\sum_{s_c' \in S}\exp(\varphi(s_c')\cdot\theta(w))}, \quad (2)$$

where $\varphi(\cdot)$ maps s_c into a vector space. Likewise, $P\big(w_c\big|\theta(w)\big)$, $P\big(s_c\big|\theta(s)\big)$, and $P\big(w_c\big|\theta(s)\big)$ are modeled in the same way. Finally, $\theta(\cdot)$ and $\varphi(\cdot)$ is optimized by maximizing Equation (1).

3.3 Training

There are $2 \times (\mathrm{N} + \mathrm{M}) \times \mathrm{D}$ parameters, including $\theta(\cdot)$ and $\varphi(\cdot)$, to be optimized, where N is the number of songs, M is the number of words, and D is the dimension of the representation. The parameters are optimized by maximizing Equation (1) using the Adam algorithm [25]. However, the computation cost of the optimization is proportional to N and M because of the normalization term in the softmax function. As an alternative, we adopt the negative sampling approach [24] to reduce the computational cost, where 30 negative pairs are randomly sampled for each positive pair.

In our experiments, the dimension of the representation was set to 32, and the hyper-parameters of the Adam algorithm were $\alpha = 0.025$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1e^{-08}$. The training was repeated for five epochs.

3.4 Text-based Music Retrieval

The response to a text query $q \in W$ is the songs that are the nearest neighbors of q in the latent space. Specifically, the cosine similarity between each $s_n \in S$ and q is calculated:

$$cosine\ simiarity = \frac{\theta(s_n)\cdot\theta(q)}{\|\theta(s_n)\|_2\|\theta(q)\|_2}, \quad (3)$$

where $\|\cdot\|_2$ denotes the Euclidean norm of a vector. The songs having high cosine similarity are the response to q.

4. EXPERIMENTS

In this section, we describe the experiments conducted to evaluate the performance of the proposed method against matrix factorization, which is another typical approach to representation learning. We first describe the dataset used in the experiments and the pre-processing step applied to the dataset. Then, we describe the implementation details of matrix factorization. Finally, we describe the results of performance evaluation.

4.1 Dataset and Pre-processing

The dataset was collected by Pichl et al. [3] using Spotify API[1]. It contains 21,485 playlists created by 1,500 users, and each playlist contains a title and a list of songs. Standard natural language processing techniques were applied to process the playlist titles. First, all characters in playlist titles were converted to lowercase, and punctuations and stop words, such as “the”, “of”, and “a”, were removed. Then, each playlist title was segmented into a set of words using the NLTK toolkit[2], and single

[1] https://developer.spotify.com/web-api/

[2] http://www.nltk.org/

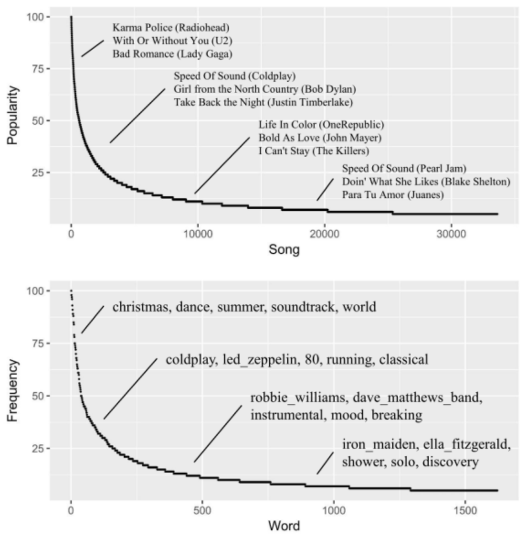


Figure 2. Song popularity and word frequency.

characters or digits were removed from the resulting word set. As many playlists titles contain artist names, the name entity recognition implemented in the NLTK toolkit was applied to identify artist names. Such artist names were considered one entity. The space in an artist name was replaced with the symbol "_" for the convenience of data processing. As shown in Table 1, the words extracted from the playlist title "George Michael - For the Heart" are "george_michael" and "heart".

However, like in other popularity studies [26], we found a serious long tail phenomenon: Few songs and words appear frequently while many others appear rarely. Both kinds of songs and words may affect representation learning. Therefore, we removed songs and words that appear less than 4 times or more than 100 times in the dataset. Interestingly, similar to stop words, words like "radio", "liked", and "music" are not useful for music retrieval but they were automatically removed because they appear many times in playlist titles. At the end of this filtering processing, 33,625 songs and 1,623 words were left (a playlist was removed if its songs and words were all removed). The statistics of the final dataset is listed in Table 2, and the song popularity (the number of times a song appears in playlists) and word frequency (the number of times a word appears in playlist titles) are shown in Figure 2. Note that the entire dataset was used for representation learning, and the performance of the representation for music retrieval was evaluated.

4.2 Matrix Factorization

Matrix factorization (MF) [14, 15] is compared with the proposed method. In MF, the vector $\boldsymbol{x}_w$ for word w and the vector $\boldsymbol{y}_s$ for song s are learned by solving the optimization problem

$$\min_{q_*,p_*} \sum_{w,s}(c_{ws} - \boldsymbol{x}_w^T \boldsymbol{y}_s)^2 + \lambda(\|\boldsymbol{x}_w\|^2 + \|\boldsymbol{y}_s\|^2), \quad (4)$$

where c_{ws} is the number of times w and s co-occur in the playlists, and λ is a regularization parameter to avoid overfitting. The inner product of a query vector and each song vector is calculated to determine which music piece to retrieve. A song with a higher inner product value is considered a better response to the query.

Number of playlists	18,417
Number of songs	33,625
Number of words	1,623
Average number of songs per playlist	20.37
Average number of words per playlist	1.10

Table 2. Data statistics.

We adopted the implementation by MyMediaLite[3]. The dimension of the vectors learned by MF was set to 32, and λ was set to 0.015.

4.3 Performance Evaluation

We measure the quality of the response to a text query by the following five metrics:

Precision and recall: We use these two standard performance evaluation metrics to measure the relevance of a response to a query as follows:

$$precision = \frac{|S_r \cap S_t|}{|S_r|}, \quad (5)$$

$$recall = \frac{|S_r \cap S_t|}{|S_t|}, \quad (6)$$

where S_r is the set of retrieved songs (the songs in the response) and S_t is the set of relevant songs (the songs in the playlists that have the query in the titles). A high precision means that most of the retrieved songs are relevant, and a high recall means that most relevant songs are retrieved.

Coherence: This metric measures the coherence of the songs in the response to a query. Specifically, we obtain social tags of songs from Allmusic[4] and calculate pointwise mutual information (PMI) for every pair of the songs in a response. The coherence is defined as the average of the PMIs,

$$coherence = \frac{1}{\mathrm{L}} \sum_{i<j} \log \frac{P(s_i, s_j)}{P(s_i)P(s_j)}, \quad (7)$$

where L is the number of the song pairs, $P(s)$ denotes the probability of s having tags and $P(s_i, s_j)$ denotes the probability of s_i and s_j having the same tags. The coherence would be high if the songs in the response have the same social tags.

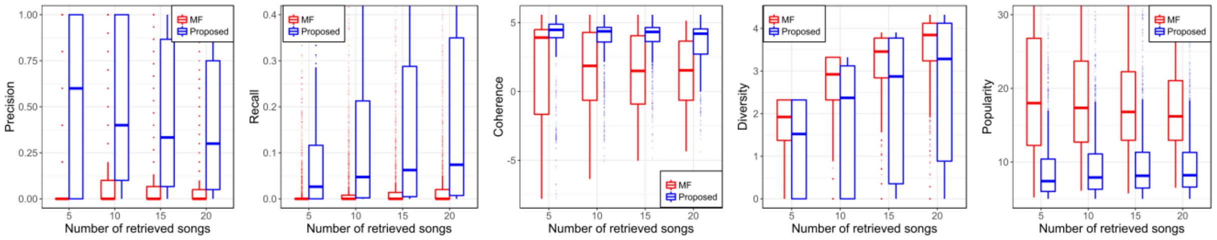
Diversity: This metric measures how diverse the songs in a response are [4, 27]. The diversity is defined as the cross entropy of artists appearing in the response:

$$diversity = \sum_{a \in A} P(a) \log(P(a)), \quad (8)$$

where A represents the set of artists in the response, and $P(a)$ denotes the probability of artist a appearing in the response. The diversity would be high if various artists appear in the response.

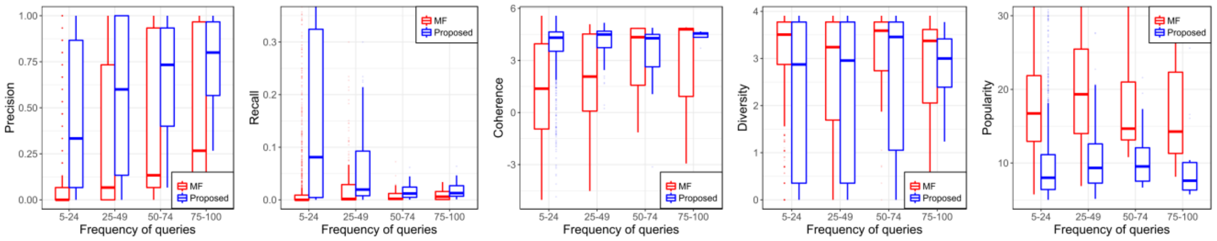
[3] http://www.mymedialite.net/

[4] http://www.allmusic.com/discover



(a) (b) (c) (d) (e)

Figure 3. Performance comparison of the proposed method and MF. The results are shown as box plots [28], where the bottom and top of a box are the first and third quartiles, and the band inside the box is the second quartile (the median). Please refer to [28] for the details of box plot.



(a) (b) (c) (d) (e)

Figure 4. Performance comparison of the proposed method and MF for queries with different word frequency. The number of retrieved songs is set to 15.

Popularity: We calculate the average popularity of the songs in a response [27],

$$popularity = \frac{1}{\mathrm{K}} \sum_{k \le \mathrm{K}} PS_k, \quad (9)$$

where K is the number of songs in the response, and PS_k represents how many times a song appears in the dataset. A low-popularity response is desired for a music retrieval and recommendation system because users may discover songs they have never heard before.

The response quality of the proposed method is compared with that of MF by the five metrics. Every word in the vocabulary (W) is considered a query to retrieve k relevant songs using a method. The average results are shown in Figure 3. We can see from Figures 3(a) and 3(b) that the precision and recall of the proposed method are higher than those of MF. It shows the effectiveness of the proposed method for preserving the relevance between songs and words. In Figure 3(c), we can see that the proposed method outperforms MF in terms of coherence. As more songs are retrieved, our method provides a stable coherence while MF has a descending coherence. In Figure 3(d), we can see that our method has a large variation in terms of diversity. It is because many responses of our method contain only the songs of one artist (zero diversity) if the query is an artist name. In Figure 3(e), we can see that the responses of the proposed method tend to have lower popularity than the responses of MF. It implies that MF favors popular songs.

We compare the responses of queries with different word frequencies. As shown in Figure 4, queries are divided into four groups according to the word frequency. We can see from Figures 4(a) and 4(b) that the proposed method has higher precision and recall than MF for each group. We can also see from Figure 4(c) that our method provides a stable coherence regardless of the frequency of query words while MF favors queries with high word frequency. In Figure 4(d), it is interesting to see that our method yields a low diversity for some queries with low word frequency. It is because part of the words with low frequency are artist names. Figure 4(e) shows that the proposed method provides responses with low popularity regardless of the word frequency of queries.

4.4 Qualitative Study

We show the responses of the two methods under comparison to five queries (“christmas”, “punk”, “60s”, “coldplay”, and “miles_davis”) in Table 3. The five queries are selected manually to cover various semantic meanings and word frequencies. Additional results and visualization of the learned latent space are provided on our website[5].

The first query “christmas” has a high word frequency, which means this word is frequently used in playlist titles. We can see that both the proposed method and MF can find songs relevant to Christmas. However, we note that the response of MF contains only two artists (actually, four of the five songs in Table 3 belong to the same artist) and has a high popularity. In contract, our method can find songs with high diversity and low popularity. The second query “punk” has a lower word frequency than the first query. We can see that the proposed method still provides a good response, while the response of MF is not very relevant to “punk”. It implies that MF may fail when the query has a low word frequency.

[5] http://mpac.ee.ntu.edu.tw/chiahaochung/textMR.php

Query	Matrix factorization	Proposed method
christmas (98[a])	It's Beginning To Look A Lot Like Christmas (Michael Bublé[b], 38[c]) All I Want For Christmas Is You (Mariah Carey, 40) White Christmas (Michael Bublé, 23) Santa Claus Is Coming To Town (Michael Bublé, 15) All I Want For Christmas Is You (Michael Bublé, 25)	Queen Of The Winter Night (Trans-Siberian Orchestra, 5) O Come All Ye Faithful/ O Holy Night (Trans-Siberian Orchestra, 6) Rudolph The Red Nosed Reindeer (Burl Ives, 6) Rockin' Around The Christmas Tree (She & Him, 5) Christmas Is Going To The Dogs (Eels, 6)
punk (23)	Sing (Ed Sheeran, 68) Shirtsleeves (Ed Sheeran, 17) Don't Let It Go (Beck, 30) Somewhereinamerica (JAY Z, 24) Bloodstream (Ed Sheeran, 29)	I Want To Conquer The World (Bad Religion, 6) Story of My Life (Social Distortion, 19) Monosyllabic Girl (NOFX, 6) Generator (Bad Religion, 10) Leave It Alone (NOFX, 8)
60s (13)	Together (Calvin Harris, 8) The Card Cheat (The Clash, 6) Bowery (Local Natives, 14) You Make Loving Fun (Fleetwood Mac, 34) Second Hand News - Early Take (Fleetwood Mac, 6)	Take Good Care Of My Baby (Bobby Vee, 5) Silence Is Golden (The Tremeloes, 5) Wooly Bully (Sam The Sham & The Pharaohs, 8) Daydream (The Lovin' Spoonful, 11) Blue Velvet (Bobby Vinton, 10)
coldplay (40)	Charlie Brown (Coldplay, 51) Major Minus (Coldplay, 17) Mylo Xyloto (Coldplay, 19) Hurts Like Heaven (Coldplay, 36) Every Teardrop Is a Waterfall (Coldplay, 42)	U.F.O. (Coldplay, 19) Prospekt's March/Poppyfields (Coldplay, 12) White Shadows (Coldplay, 15) Mylo Xyloto - Live (Coldplay, 7) Twisted Logic (Coldplay, 9)
miles_davis (7)	Scarborough Fair / Canticle (Simon & Garfunkel, 14) Is She Weird (Pixies, 6) Shoes Upon the Table (Blood Brothers - 1995 London Cast, 5) I Would For You (Nine Inch Nails, 13) Love Is The Answer (Aloe Blacc, 13)	Fran-Dance (Miles Davis, 7) On Green Dolphin Street (Miles Davis, 7) Spanish Key (Miles Davis, 5) Flamenco Sketches (Miles Davis, 13) Love For Sale (Miles Davis, 8)

Table 3. Qualitative Study. Only the top five songs to a query are shown. ([a] word frequency, [b] artist, [c] song popularity)

The query "60s" is interesting, as it is related to the songs or artists in 1960s. We can see that our method can find the songs of artists who were popular in 1960s, including Bobby Vee, The Tremeloes, Sam The Sham & The Pharaohs, The Lovin' Spoonful, and Bobby Vinton. MF fails in this case because "60s" has a low frequency.

The last two queries "coldplay" and "miles_davis" are both artists, where the former has higher word frequency than the latter. We can see that our method provides good responses to the two queries, while MF fails in the case of "miles_davis". It can be expected that the response to this kind of query should contain only the songs of the artist specified in the query. Note that there are many artist names in our vocabulary, and most of them have low word frequency. Because the proposed method works for these artist queries as well as other queries, the diversity of the proposed method has a large variation, as shown Figures 3(d) and 4(d).

5. DISCUSSION

We first discuss the difference between the proposed method and MF in terms of the learning function. As described in Equation (4), MF considers only the co-occurrence of song-word pairs. In contrast, our method exploits three types of co-occurrence between songs and words of playlists. Although an improved MF [29] can be applied to factorize multiple co-occurrence matrices, we believe that the property of MF (i.e. the favor of popular songs and words) would make MF unsuitable for text-based music retrieval.

Our method is related to the embedding method proposed by Moore et al. [20] for representation learning of songs and tags for playlist prediction. The difference between our method and their method lies in the function used to model the conditional probability: a softmax function vs a logistic function that uses the Euclidean distance between two vectors as input. Besides, we applied two modern approaches, the Adam algorithm [25] and the negative sampling [24], to improve the efficiency of representation learning.

Finally, we discuss two possible directions to extend the proposed method. One direction is to enlarge song set and word set. As song titles and lyrics also contain rich text information, they can be incorporated to expand word set. Besides, the approach proposed by Oord et al. [30] can be applied to map new songs into the latent space learned by our method. This approach also solves the cold start problem [27]. The other direction is to develop a music retrieval system which allows multiple words as a query, because people may use multiple words or even a sentence to retrieve music. There are simple solutions, for example, combining the responses to multiple single-word queries [6]. However, such combination may not truly capture the semantic meaning of a multiple-words query. To deal with such query, a better solution, such as the approach proposed by Mikolov et al. [24], can be incorporated into the proposed method. We can see the potential and high extendability of our method.

6. CONCLUSION

In this paper, we have proposed an unsupervised learning method to generate the latent representation of songs and words of playlists for text-based music retrieval. Such representation captures the relevance between songs and words, owning to the proximity property of the latent space. Both qualitative and quantitative evaluations show the effectiveness of the proposed method compared against the matrix factorization method for text-based music retrieval.

7. REFERENCES

[1] A. N. Hagen, “The playlist experience: Personal playlists in music streaming services,” *Popular Music and Society*, vol. 38, no. 5, pp. 625–645, 2015.

[2] S. J. Cunningham, D. Bainbridge, and A. Falconer, “More of an art than a science: Supporting the creation of playlists and mixes,” in *Proc. 7th Int. Conf. Music Inf. Retrieval (ISMIR)*, pp. 240–245, 2006.

[3] M. Pichl, E. Zangerle, and G. Specht, “Towards a context-aware music recommendation approach: What is hidden in the playlist name?” in *Proc. 15th IEEE Int. Conf. Data Mining Workshop (ICDMW)*, pp. 1360–1365, 2015.

[4] B. Fields, “Contextualize your listening: The playlist as recommendation engine,” PhD dissertation, Dept. Comput., Goldsmiths, Univ. London, 2011.

[5] D. Jannach, I. Kamehkhosh, and G. Bonnin, “Analyzing the characteristics of shared playlists for music recommendation,” in *Proc. 6th Workshop Recommender Syst. Social Web (RSWeb)*, 2014.

[6] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet, “Towards musical query-by-semantic-description using the CAL500 data set,” in *Proc. 30th Int. ACM Conf. Res. Develop. Inf. Retrieval (SIGIR)*, pp. 23–27, 2007.

[7] G. Chechik, E. Ie, M. Rehn, S. Bengio, and D. Lyon, “Large-scale content-based audio retrieval from text queries,” in *Proc. 1st ACM Int. Conf. Multimedia Inf. Retrieval*, pp. 105–112, 2008.

[8] N. Hariri, B. Mobasher, and R. Burke, “Personalized text-based music retrieval,” in *Workshops 27th AAAI Conf. Artificial Intell.*, 2013.

[9] Z. Cheng, J. Shen, and S. C.H. Hoi, “On effective personalized music retrieval by exploring online user behaviors,” in *Proc. 39th Int. ACM Conf. Res. Develop. Inf. Retrieval (SIGIR)*, pp. 125–134, 2016.

[10] P. Knees, T. Pohle, M. Schedl, D. Schnitzer, and K. Seyerlehner, “A document-centered approach to a natural language music search engine,” in *European Conf. Inf. Retrieval (ECIR)*, Springer Berlin Heidelberg, pp. 627–631, 2008.

[11] P. Knees, T. Pohle, M. Schedl, D. Schnitzer, K. Seyerlehner, and G. Widmer, “Augmenting text-based music retrieval with audio similarity,” in *Proc. 10th Int. Soc. Music Inf. Retrieval Conf. (ISMIR)*, pp. 579–584, 2009.

[12] P. Knees, M. Schedl, T. Pohle, K. Seyerlehner, and G. Widmer, “Supervised and unsupervised web document filtering techniques to improve text-based music retrieval,” in *Proc. 11th Int. Soc. Music Inf. Retrieval Conf. (ISMIR)*, pp. 543–548, 2010.

[13] G. Dror, N. Koenigstein, and Y. Koren, “Yahoo! music recommendations: Modeling music ratings with temporal dynamics and item taxonomy,” in *Proc. 5th ACM Int. Conf. Recommender Syst. (RecSys)*, pp. 165–172, 2011.

[14] Y. Koren, R. Bell, and C. Volinsky, “Matrix factorization techniques for recommender systems,” *IEEE Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[15] Y. Hu, Y. Koren, and C. Volinsky, “Collaborative filtering for implicit feedback datasets,” in *Proc. 8th IEEE Int. Conf. Data Mining (ICDM)*, pp. 263–272, 2008.

[16] C.-M. Chen, M.-F. Tsai, Y.-C. Lin, and Y.-H. Yang, “Query-based music recommendations via preference embedding,” in *Proc. 10th ACM Conf. Recommender Syst. (RecSys)*, pp. 79–82, 2016.

[17] C.-M. Chen, C.-Y. Yang, C.-C. Hsia, Y. Chen, and M.-F. Tsai, “Music playlist recommendation via preference embedding,” in *Poster Proc. 10th ACM Conf. Recommender Syst. (RecSys)*, 2016.

[18] J. Weston, S. Bengio, and P. Hamel, “Large-scale music annotation and retrieval: Learning to rank in joint semantic spaces,” *arXiv preprint arXiv: 1105.5196*, 2011.

[19] S. Chen, J. L. Moore, D. Turnbull, and T. Joachims, “Playlist prediction via metric embedding,” in *Proc. 18th ACM Int. Conf. Knowledge Discovery Data Mining (SIGKDD)*, pp. 714–722, 2012.

[20] J. L. Moore, S. Chen, T. Joachims, and D. Turnbull, “Learning to embed songs and tags for playlist prediction,” in *Proc. 13th Int. Soc. Music Inf. Retrieval Conf. (ISMIR)*, pp. 349–354, 2012.

[21] J. L. Moore, S. Chen, T. Joachims, and D. Turnbull, “Taste over time: The temporal dynamics of user preferences,” in *Proc. 14th Int. Soc. Music Inf. Retrieval Conf. (ISMIR)*, pp. 401–406, 2013.

[22] C.-H. Chung, J.-K. Lou, and H. Chen, “A latent representation of users, sessions, and songs for listening behavior analysis,” in *Proc. 17th Int. Soc. Music Inf. Retrieval Conf. (ISMIR)*, 2016.

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

[24] T. Mikolov, I. Sutskever, K. Chen,G. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Proc. Advances in Neural Inf. Process. Syst. (NIPS)*, pp. 3111–3119, 2013.

[25] D. P. Kingma and J. L. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv: 1412.6980*, 2014.

[26] O. Celma, *Music Recommendation and Discovery*, Springer Berlin Heidelberg, 2010.

[27] S.-Y. Chou, Y.-H. Yang, and Y.C. Lin, “Evaluating music recommendation in a real-world setting: On data splitting and evaluation metrics,” in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, pp. 1–6, 2015.

[28] R. McGill, J. W. Tukey and W. A Larsen, “Variations of box plots,” *American Statistician*, vol. 32, no. 1, pp.12–16, 1978.

[29] A. Vall, M. Skowron, P. Knees, and M. Schedl, “Improving music recommendations with a weighted factorization of the tagging activity,” in *Proc. 16th Int. Soc. Music Inf. Retrieval Conf. (ISMIR)*, pp. 65–71, 2015.

[30] A. Van de Oord, S. Dieleman, and B. Schrauwen, “Deep content-based music recommendation,” in *Proc. Advances Neural Inform. Process. Syst. (NIPS)*, pp. 2643–2651, 2013.