

# Visualization of the Artist Relations Using Twitter User Profiles

Minoru YOSHIDA <sup>a,1</sup>, Shogo KOHNO <sup>a</sup> and Kazuyuki MATSUMOTO <sup>a</sup> and Kenji KITA <sup>a</sup>  
<sup>a</sup>*Tokushima University*

**Abstract.** We propose a new music artist recommendation algorithm using *Twitter profile texts*. Today, *music recommendation* is provided in many music streaming services. In this paper, we propose a new recommendation algorithm for this music recommendation task. Our idea is to use *Twitter profile texts* to find appropriate artist names to recommend. We obtained word embedding vectors for each artist name by applying word2vec algorithm to the corpus obtained by collecting such user profile texts, resulting in vectors that reflect artist co-occurrence in the profile texts.

**Keywords.** music recommendation, Twitter profiles

## 1. Introduction

We propose a new music artist recommendation algorithm using *Twitter profile texts*.

Today, many songs can be accessed on the Internet. In many cases, users access to these songs by the artist names, e.g., search for songs by inputting the artist name as keyword to a search box. In this scenario, users can access only to songs performed by the artist input to the system. This approach has its merits such as users can access to desired songs with high certainty (i.e., high accuracy), but its demerits such as the user is unlikely to find unpredicted but good (i.e., fit to the user's taste) songs. To cover the latter demerits of simple keyword search, *music recommendation* is provided in many music streaming services. Typically, such recommendation provides a list of songs that are similar to the song currently listened to by the user.

In this paper, we propose a new recommendation algorithm for this music recommendation task. Our idea is to use *Twitter profile texts* to find appropriate artist names to recommend. We assume that many Twitter users list their favorite things in their social networking services (SNS) profile texts and we can thus find the artist who is likely to be preferred by the fans of artist *a* by looking at the Twitter profile text of the fans of *a*.

We obtain word embedding vectors for each artist name by applying word2vec algorithm to the corpus obtained by collecting the user profile texts, resulting in vectors that reflect artist co-occurrence in the profile texts. Our recommendation system also has potential to assist the simple artist name search approach, e.g., even if the user does not know the artist name of the song *s*/he is searching for, *s*/he can guess the (potentially

---

<sup>1</sup>Tokushima University; E-mail: mino@is.tokushima-u.ac.jp

wrong) artist name from the song's flavor or mood and input it to the system, then the system can suggest the list of artists which may include the correct artist name.

## 2. Related Work

Music recommendation research has been mainly studied by audio or music-IR research communities [1]. Traditional approaches to music recommendation are to use collaborative filtering approach, which makes a user-item matrix (where each entry  $e_{ij}$  represents the user- $i$ 's rating on item- $j$ ) and calculate item-item similarity based on the obtained matrix [2]. In another approach, Moling et al. [3] proposed to use a user's listening behavior on radio channels to train the agent for music recommendation. Another line of approaches is content-based recommendation using music features [2] [4]. For example, Hijikata et al. [5] used content-based filtering for music recommendation. Musical features such as tempo, rhythm, codes, timbres, keys, etc. from musical instrument digital interface (MIDI) files and used to construct music vectors, which were used to calculate similarity between songs.

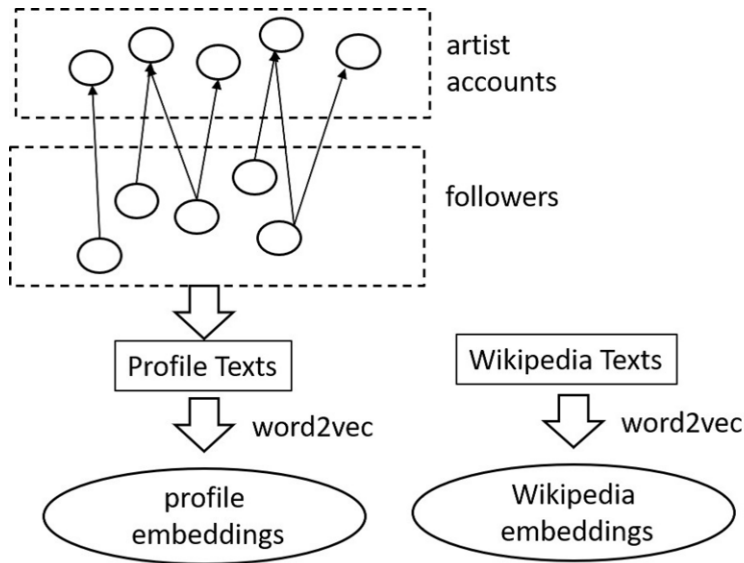
On the other hand, music recommendation using texts found on SNS have been investigated by several researchers recently. Mimura et al. [6] selected 20 music artists and collected followers of the official Twitter accounts for these artists. Tweets of these users are collected as text sets, and they applied Latent Dirichlet Allocation (LDA) to extract topics. Finally, artist similarity was defined as the topic distribution similarity. Sasai [7] proposed *ArtistVector* which is embedding vectors for music artists. He collected texts for artists from Wikipedia articles and web texts written by fans of artists, and applied several algorithms including word2vec to the texts to obtain document embedding vectors. The resulting document vectors were used as artist embeddings to visualize artist similarity. Hirota et al. [8] collected texts from Web dictionary and curation sites for each artist, and obtained artist vectors by counting the number of each word. The extracted context words were used to provide the interface to select desired contexts to be used for recommendation. Hyung et al. [9] extracted texts from song requests in radio and recommended artists based on the extracted texts.

Compared to these previous approaches, we propose a new method to obtain artist vectors that uses *Twitter profile texts*. Recently, several researchers have used Twitter user profiles for automatic labeling of Twitter users. For example, [10] used Twitter profile texts to classify users' political affiliations into "Democrats", "Republicans" or "Unknown". However, approaches that use user profiles as general text corpus have not been studied actively [11].

### 2.1. Word2vec

Word2vec is a standard tool to obtain word embedding in low (typically 50-500) dimensions. Skip-gram with Negative Sampling (SGNS) approach is frequently used to obtain such vectors. It maximizes the sum of the plausibility scores

$$\log \sigma(v'_{w_o} \cdot v_{w_l}) + \sum_{k=1}^K \log \sigma(-v'_{w_k} \cdot v_{w_l})$$



**Figure 1.** System Workflow of Our Method

of predicting the “output word”  $w_O$  given the “input word”  $w_I$ , where  $w_O$  and  $w_I$  appears in the same *context* i.e., in text window of the pre-defined size. Here,  $K$  is the number of negative samples,  $v_w$  and  $v'_w$  are two types of vectors called the “input” and “output” vectors, respectively, and  $\sigma$  is the sigmoid function. Negative samples  $w_k$  are selected randomly in each learning step.

This approach is based on the *distributional hypothesis* that assumes *words in similar meaning appear in the similar contexts* where *contexts* is the words that appear around the target word.

### 3. Problem Setting

We assume a user gives one or two artist names to the system. The system returns the names of other artists *similar* to the query artists. We consider two types of similarities: document-based similarity calculated with *Wikipedia vectors* and profile-based similarity calculated with *profile vectors* described later.

To make the corpus balanced and comprehensive, we selected 6 artist categories: “J-POP”, “Idols”, “Rock”, “Alternative Pops”, “Anime”, and “Others” and collected equal amounts of texts from each category.

### 4. Proposed Method

We observed that many users wrote two or more artist names in their profiles (e.g., a user who writes one rock artist name in his/her profile tend to write another rock artist in his/her profile,) and in many cases these names were in near positions in the text. This means that we can obtain co-occurrence statistics for artist names found in the profile

texts. Our idea is to obtain the artist embeddings by running word2vec on these profile texts because word2vec learn embedding vectors using word co-occurrences in limited-size windows.

#### 4.1. Corpus

Figure 1 shows an overview of our method. We selected twitter users who show strong interest in some music artists (i.e., we collected fans of some music artists.) More concretely, we selected users who follow one or more music artists or music events. We used TwitterAPI to obtain Twitter profiles and main texts (i.e., tweets) for the selected users.

We selected the 50 Twitter accounts of artists or events related to the above-mentioned categories. We call these selected accounts “artist accounts”. We collected around 1,000 followers for each artist account and extracted their profiles. As a result, we obtained around 50,000 accounts without considering overlapping. All profile texts from them were concatenated into one file, resulting in a 262,280-lines text<sup>2</sup>. However, we found overlap between these lines because the same person may follow two or more artist accounts. After removing overlap, we obtained the final profile texts that consisted of 32,781 lines.

The collected profile texts looks like as follows.<sup>3</sup>

I'm AKB48 fan living in Tokyo. I also love SKE48 and NMB48!

We obtained *artist embeddings* learned by running word2vec on the obtained texts.

#### 4.2. Implementation Issues

Profile texts were separated into word lists by using the morphological analyzer McCab-NEologd [12] [13]

We constructed the *artist dictionary*, which consists of lists of synonyms for each artist name. These synonyms (e.g., nicknames) were obtained mainly from the artist-name list in Uta-Map<sup>4</sup>. We also extracted some acronyms of the artist names by seeing the obtained corpus. This dictionary was used to normalize the artist names found in the corpus. We also implemented the preprocessor that remove noises like URLs and face marks.

We used word2vec[14] to obtain word embedding vectors where the parameters were set as follows:

- algorithm = skip-gram,
- dimension size = 100,
- window size = 15,
- word frequency threshold = 5.

---

<sup>2</sup>It was larger than 50,000 because we might extract multiple lines from one profile text.

<sup>3</sup>This is an imaginary example because of privacy issues.

<sup>4</sup><https://www.utamap.com/>

### 4.3. Visualization by t-SNE

Obtained similarity between artists can be used to visualize music artist relations. We implemented a system where a user give two artist names as a query, and system returns the music artist map considered as *similar* to one or two given artists. Visualization was performed by using t-SNE [15], which embeds high-dimensional vectors into 2-dimensional points where two points become near if the distance between two original high-dimension vectors is small.

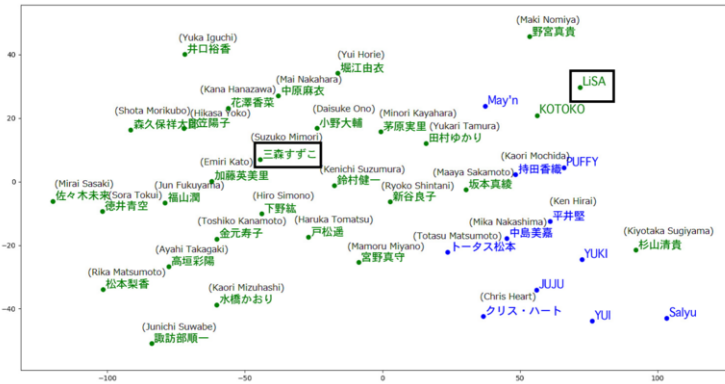


Figure 2. Example Result Using Wikipedia

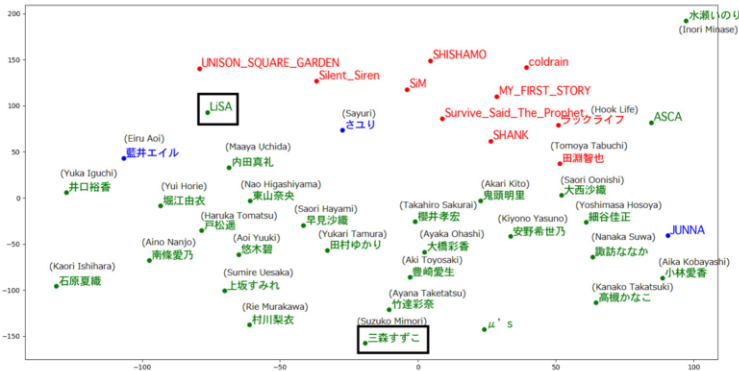


Figure 3. Example Result by Proposed Method

As described later, we tested two types of vectors: *Wikipedia vectors* learned by using Wikipedia texts and *profile vectors* learned by our proposed method. Results are shown in Figure 2 (using Wikipedia text) and 3 (using profile texts) for the query “LiSA” and “Suzuko Mimori”, both of which are Japanese Voice Actors, but the former is also a rock artist. In these figures, generally speaking, names with Kanji characters are of voice actors, and names with alphabets are for rock artists.

We observed that the results using profile texts (Figure 3) gave more comprehensive visualization where rock artists (upper half, colored red in Figure 3) were located near

“Lisa” and female voice actor names (lower half, colored green in Figure 3) were located near “Suzuko Mimori” than results using Wikipedia texts (Figure 2). In Figure 2, some voice actors and major music artists were listed (green names indicate the name of voice actors (including male and female ones) and blue names indicate the name of J-POP artists), but easy-to-understand structures were not observed. For example, voice actors listed in Figure 2 include both male and female ones, while the most of voice actors listed in Figure 3 (our method) were female voice actors which can be considered more similar to the query artists (both of them were female ones).

## 5. Experiments

We compared two types of embedding vectors: *Wikipedia vectors* provided by Shiro-Yagi Corp.<sup>5</sup> which is pretrained on Japanese Wikipedia texts, and *profile vectors* obtained by using our method which uses Twitter profiles as a corpus. We call the former *Wikipedia vectors* and the latter *profile vectors*.

We selected 10 query artists (2 for each genre excluding “others” category) for evaluation. The Twitter profile corpus was divided into a training (80%) and test(20%) set, and we re-train our model using the training set. For each query artist  $q$ , we collected the test set profile texts that included  $q$ . The task is to predict the artist names other than  $q$  included in each collected profile text. Therefore, the experiment is to test if we can predict a user’s taste by using vectors obtained using other users’ tastes.

We used *average precision* [16] for the evaluation metric. We ranked all artist names using cosine similarity of vectors of each artist and the query artist  $q$ . Given the ranked list of artists  $\langle c_1, c_2, \dots, c_n \rangle$ , we calculate average precision for each profile text that includes artist names  $S = \{q, s_1, s_2, \dots\}$  as follows:

$$\frac{1}{|S|} \sum_{1 \leq k \leq n} r_k \cdot \text{precision}(k),$$

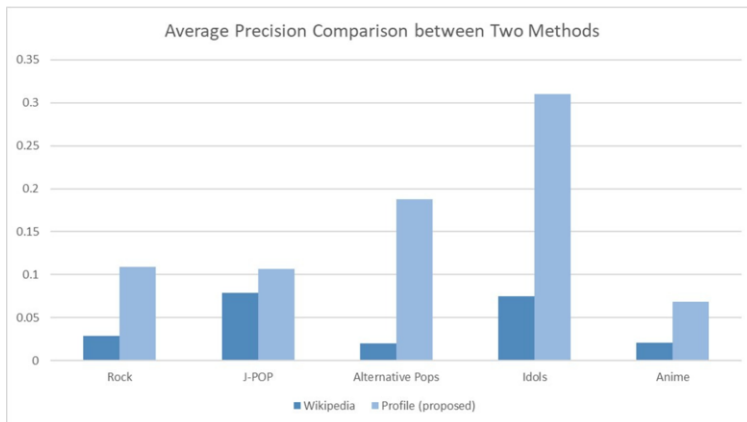
where  $\text{precision}(k)$  is the accuracy (i.e., ratio of correct answers to all answers) of the top  $k$  candidates, and  $r_k$  represents whether the  $k$ -th artist is relevant (1) or not (0). (In other words,  $r_k = 1$  if  $c_k \in S$ , and  $r_k = 0$  otherwise.)

**Table 1.** Average Precision by using profile and Wikipedia vectors

Genre	Wikipedia	Profile (proposed)
Rock	0.0287	0.109
J-POP	0.0786	0.107
Alternative Pops	0.0204	0.188
Idols	0.0749	0.310
Anime	0.0210	0.0689

Results are shown in Table 1 and Figure 4. We observed high average precision in the results by the proposed method, especially in the Idols genre, compared to the Wikipedia vectors. We think the reasons for good results in Idols genre is that this genre is relatively “exclusive” which means that the fans of an idol are likely to choose another

<sup>5</sup><https://aial.shiroyagi.co.jp/2017/02/japanese-word2vec-model-builder/>



**Figure 4.** Comparison of Average Precision between Two Methods

idol as another artist to listen to. On the other hand, for J-POP the difference between two methods was small, suggesting that J-POP genres do not have such strong exclusiveness probably because the genre is more general (i.e., contain various kinds of artists) than other genres.

## 6. Conclusions and Future Work

In this paper, we proposed a new artist recommendation algorithm using *Twitter profile texts* to reflect people's tastes on music artists more directly. We also validated our results by visualizing the list of artists obtained by using two different queries using t-SNE. We observed that the proposed method provided more discriminative results than a method using Wikipedia texts in some genres.

Future work includes hybrid approaches of profile and Wikipedia texts, which may provide profile-based taste-oriented recommendation as well as Wikipedia-based exhaustive recommendations simultaneously. Developing systems with integrated ranking algorithms that consider not only artist similarities but also song similarities or genre similarities in one scoring function is also important future work.

## Acknowledgement

This work was supported by JSPS KAKENHI Grant Numbers JP18K11549, JP20K12027.

## References

- [1] Jannach D, Kamehkhosh I, Bonnin G. Music Recommendations: Algorithms, Practical Challenges and Applications. In: Berkovsky S, Cantador I, Tikk D, editors. Collaborative Recommendations: Algorithms, Practical Challenges and Applications. World Scientific; 2018. p.481-518.
- [2] Yoshii K, Goto M, Komatani K, Ogata T, Okuno HG. Hybrid collaborative and content-based music recommendation using probabilistic model with latent user preferences. In: Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR); 2006. p. 296-301.

- [3] Moling O, Baltrunas L, Ricci F. Optimal radio channel recommendations with explicit and implicit feedback. In: Proceedings of the sixth ACM conference on Recommender systems (RecSys '12); 2012. p. 75-82.
- [4] Hartono P, Yoshitake R. Automatic Playlist Generation from Self-Organizing Music Map. *Journal of Signal Processing*. 2013; 17(1):11-19.
- [5] Hijikata Y, Iwahama K, Nishida S. Content - based Music Filtering System using Decision Tree and its Evaluation. *IPSJ SIG Technical Reports*. 2004; 2004-DBS-133:17-24. (In Japanese).
- [6] Mimura N, Ushiyama T. Can we see the fans from SNS? - Extraction of fan characteristics of music artists using Twitter -. In: Proceedings of the DEIM Forum; 2017. D8-1. (In Japanese).
- [7] Sasai D. ArtistVector: Artist Feature Extraction with Web Document Embeddings. *IPSJ Technical Report*. 2018; 2018-NL-236(3):1-7. (In Japanese).
- [8] Hirota M, Okada R, Nakanishi T, Kitagawa T. Dynamic Context-Dependent Artist Network Generation for Japanese Music Artists. In: Proceedings of the DEIM Forum; 2016. E1-1. (In Japanese).
- [9] Hyung Z, Lee K, Lee K. Music recommendation using text analysis on song requests to radio stations. *Expert Systems with Applications*. 2014; 41(5):2608-2618.
- [10] Joshi A, Bhattacharyya P, Carman MJ. Political Issue Extraction Model: A Novel Hierarchical Topic Model That Uses Tweets By Political And Non-Political Authors. In: Proceedings of WASSA, NAACL-HLT; 2016. p. 82-90.
- [11] Yoshida M, Matsumoto K, Kita K. Modeling Relations Between Profiles and Texts. In: Proceedings of AIRS 2018; 2018. p. 103-109.
- [12] Kudo T. MeCab:Yet Another Part-of-Speech and Morphological Analyzer, "<http://taku910.github.io/mecab/>", 2013.
- [13] Kudo T, Yamamoto K, Matsumoto Y. Applying Conditional Random Fields to Japanese Morphological Analysis. In: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP-2004); 2004. p. 230-237.
- [14] Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J. Distributed representations of words and phrases and their compositionality. In: Proceedings of the Advances in neural information processing systems; 2013. p. 3111-3119.
- [15] Maaten LVD, Hinton G. Visualizing data using t-SNE. *Journal of Machine Learning Research*. 2008 Nov;9:2579-2605.
- [16] Chakrabarti S. *Mining the Web : Discovering Knowledge from Hypertext Data*. Morgan-Kaufmann Publishers; 2002. 368 p.