COMBINING MUSIC SPECIFIC EMBEDDINGS FOR COMPUTING ARTIST SIMILARITY

Pasquale Lisena
EURECOM, Sophia Antipolis, France
pasquale.lisena@eurecom.fr

Raphaël Troncy
EURECOM, Sophia Antipolis, France
raphael.troncy@eurecom.fr

ABSTRACT

In this paper, we present an original approach and preliminary results for computing similarities between music-related entities (artists, works, performances) using embeddings that take into account not only the semantic description of those entities but also their usage in a rich music dataset.

1. INTRODUCTION

An artist, and in particular a musician, can be described using different features: as a person, one can use the date and place of birth and death; as a composer, once can consider the musical genre, the foreseen medium of performance (MoP), the key, *etc.*. of the compositions; as a performer, one can think of the function and role of the artist, the actual MoP played, *etc.*. during the performance. This information is representative of the career of an artist. We hypothesis that this kind of metadata plays an important role when one has to compute similarity between artists [4, 6].

The DOREMUS dataset [1] publicly exposes this kind of information as linked data through a SPARQL endpoint [1]. Alongside data about artists, works and performances, it includes a set of controlled vocabularies for keys, MoPs and genres, that define labels, definitions, hierarchies and other connections between over 9,000 entities.

The distance between the embedding vectors in the Euclidean space is often used as similarity measure in the musical domain [2, 7], that have applications in music recommendation and playlists prediction.

This work introduces some preliminary results about a new strategy for computing the similarity between two artists. This strategy relies on the combination of partial embeddings, computed at the level of the single features and concatenated in order to realise the artist embedding. This approach has several advantages: we can give weights to each feature and we can fully exploit the knowledge base. Moreover, we do not have to recompute the whole embedding for new artists, skipping in this case the most computationally expensive task.

2. APPROACH

2.1 Feature Embeddings

We select the subset of features coming from the DOREMUS dataset (key, genre, MoP) for which their values are controlled by specific vocabularies, and in particular, when terms are defined with narrower-broader relationships. For each term, we compute an embedding applying *node2vec* [3] on two sub-graphs: the one of the controlled vocabularies and the one corresponding to the usage of their values in the DOREMUS dataset. A L2 normalisation is then applied in order to have values in {-1;+1}. One of the advantages of having feature-specific vectors is the possibility of reusing them in different contexts (i.e. similarity of composer, performers, works, *etc.*).

2.2 Embeddings Combination

The generation of the artist embedding is realised through the combination of its corresponding feature embedding. This step is driven by a configuration file that contains a series of properties and, for each of them, a query for the SPARQL endpoint (in which the artist is a parameter) and, eventually, the embedding vector to rely on.

For each feature and for each result of the query, we get:

- if the result is an entity, the relative vector;
- if the result is a date or a number, the same number mapped in {-1;+1};
- if there are no results (i.e. unknown values or not applicable property to particular works), an array of null values of the same length as it should be.

The mean of the different results belonging to the same feature is then computed, in order to have a unique vector. The artist vector is realised through the concatenation of each feature mean vector.

2.3 Similarity Score

We want to compute a similarity score between a seed artist s and a target one t. As seen in Section 2.1, their vectors can contains some null dimensions, which we do not want to consider. Therefore, we remove from both vectors all the dimensions that are null for any of them. On this shorter version of the vectors, we compute the squared euclidean

[1] http://data.doremus.org

Attribution: Pasquale Lisena, Raphaël Troncy. "Combining Music Specific Embeddings for Computing Artist Similarity", Extended abstracts for the Late-Breaking Demo Session of the 18th International Society for Music Information Retrieval Conference, Suzhou, China, 2017.

Artist 1	Artist 2	Score	MoP	Genre	Key	Birth dates	Death dates
A. Vivaldi	T. Albinoni	0.999985	continuo string orchestra violin	concert 17th century sonata	F Bb g	1678 - 1671	1741 - 1750
W. A. Mozart	L. v. Beethoven	0.999973	voice piano full orchestra	concert sonata quartet	Eb C D	1956 - 1770	1791 - 1827
J. Coltrane	J. Kern	0.999972	–	jazz trio	Ab Eb	1926 - 1885	1967 - 1945

Table 1. Artist similarity explanation examples.

distance d. Defining d_{max} as the distance between an all-ones vector and its additive inverse one, we compute the score according to the following formula:

$$similarity = \frac{d_{max} - d}{\mid d_{max} \mid} * (1 - penalty)$$

where the $penalty$ is the percentage of the dimensions missing in t but present in s among all the properties in s.

3. SIMILARITY EXPLANATION

With the double goal of having a feedback about the similarity score and of producing a visualisation for user interfaces, we are extracting the most similar feature values between two artists. For each feature, we perform the same queries contained in the configuration file described in Section 2.2. Then, we compare the results and select the ones that have highest similarity (i.e. same or most similar genres), giving priority to the most present ones. Table 2.3 shows some examples of two classical composer couples (for which we have more data) and a jazz one.

We integrate this implementation within the OVERTURE [2] application [5] that provides a user friendly interface to browse and explore the DOREMUS dataset.

4. CONCLUSION AND FUTURE WORK

In this paper, we present some preliminary results for computing similarity between music-related entities (artists, works, performances) using embeddings that take into account both the static description of those entities as well as usage in musical datasets.

As future work, we aim to further enrich the DOREMUS dataset with new data. The feature embeddings of the properties will then be improved, for example, using "same as" links between the different controlled vocabularies. New embeddings will be produced for functions, responsibilities and places, after the integration of this vocabularies in the dataset. Next, we plan to experiment with neural networks in order to assign different weights to the contribution of each dimension in the similarity score. We will also follow a similar approach to compute similarities not only between artists, but also between two works, two expressions or two concerts. Finally, we aim to work on dimensionality reduction in order to solve the problem of the curse of dimensionality (that has so far no relevant effects because of a limitation in the dimension at the feature embeddings level), optimising in the same time the computation of similarity so that it becomes fast enough to be included in a recommender system for works and playlists.

Acknowledgement. This work has been partially supported by the French National Research Agency (ANR) within the DOREMUS Project, under grant number ANR-14-CE24-0020.

5. REFERENCES

[1] Manel Achichi, Rodolphe Bailly, Cécile Cecconi, Marie Destandau, Konstantin Todorov, and Raphaël Troncy. DOREMUS: Doing Reusable Musical Data. In *14^{th} International Semantic Web Conference (ISWC)*, 2015.

[2] L. Espinosa-Anke, S. Oramas, H. Saggion, and Xavier Serra. Elmdist: A vector space model with words and musicbrainz entities. In *Workshop on Semantic Deep Learning (SemDeep)*, 2017.

[3] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22^{nd} ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

[4] Yang-Hui He. A Visualization of the Classical Musical Tradition. *ArXiv e-prints*, 2017.

[5] Pasquale Lisena, Manel Achichi, Eva Fernandez, Konstantin Todorov, and Raphaël Troncy. Exploring Linked Classical Music Catalogs with OVERTURE. In *15^{th} International Semantic Web Conference (ISWC)*, 2016.

[6] B. McFee and G. R. G. Lanckriet. Heterogeneous embedding for subjective artist similarity. In *10^{th} International Symposium for Music Information Retrieval (ISMIR))*, October 2009.

[7] Joshua L. Moore, Shuo Chen, Thorsten Joachims, and Douglas Turnbull. Learning to embed songs and tags for playlist prediction. In *13^{th} International Society for Music Information Retrieval Conference (ISMIR)*, pages 349–354, 2012.

[2] http://overture.doremus.org/artist