Computing the Semantic Relatedness of Music Genres using Semantic Web Data

Dennis Diefenbach
Université de Lyon, CNRS
UMR 5516 Laboratoire Hubert Curien
Saint-Etienne, France
dennis.diefenbach@univ-st-etienne.fr

Pierre-René Lhérisson
Université de Lyon, CNRS
UMR 5516 Laboratoire Hubert Curien
Saint-Etienne, France
pr.lherisson@univ-st-etienne.fr

Fabrice Muhlenbach
Université de Lyon, CNRS
UMR 5516 Laboratoire Hubert Curien
Saint-Etienne, France
fabrice.muhlenbach@univ-st-etienne.fr

Pierre Maret
Université de Lyon, CNRS
UMR 5516 Laboratoire Hubert Curien
Saint-Etienne, France
pierre.maret@univ-st-etienne.fr

ABSTRACT

Computing the semantic relatedness between two entities has many applications domains. In this paper, we show a new way to compute the semantic relatedness between two resources using semantic web data. Moreover, we show how this measure can be used to compute the semantic relatedness between music genres which can be used for music recommendation systems.

We first describe how to build a vector representations for resources in an ontology. Subsequently we show how these vector representations can be used to compute the semantic relatedness of two resources. Finally, as an application, we show that our measure can be used to compute the semantic relatedness of music genres.

CCS Concepts

•**Information systems** → **Similarity measures;** *Language models;*

Keywords

Vector Representations, Semantic Relatedness, Music Genres, Recommendation Systems

1. INTRODUCTION

The work presented in this paper is the result of a research work on music recommendation systems conducted in collaboration with a french startup company called 1D Lab[1]. The aim is to create a music recommendation system over a database of independent artists, i.e. artists who are not under contract with the major labels. The independent, emerging artists, suffer from a lack of visibility on the web. To make reccomandations we therefore have to relay on the few information we have about their tracks, like their music genre. Moreover we face the so-called cold-start problem, i.e. we cannot use user profiles to generate the recommendations.

While it is nearly impossible to find information about independent artists in the Semantic Web it is easy to find information about the music genre. These include songs of a particular genre, bands that play the genre, characteristic instruments for the genre, influences, regional distributions and many more.

The problem we address here is how to exploit semantic web data to answer question like "How similar is Ska and Reggae?" or "How similar is Ska and Classical Music?". The answer to these questions can then be used to recommend to a person that is listening Ska a song of a different but close music genre.

Answering the above questions is equivalent to find out what is the semantic relatedness between two music genres. We propose first a way to construct for resource of an ontology a vector representation of it. Using these vector representations we propose a new way to compute the semantic relatedness of two resources in an ontology. Finally, we show the results we get if we apply it to music genres.

The publication is organized as follows. In section 2 we discuss the related work. In section 3 we describe how to compute the semantic relatedness between two resources in an ontology. In section 4 we show our results when applied to music genres and finally in section 5 we discuss future work.

2. RELATED WORK

Recommender Systems are designed to guide users through large volumes of data. One of the most popular methods is collaborative-filtering where the recommendation is based on users' preferences and behaviors [10]. Another approach is content-based recommendation which uses information associated to items like unstructured text for web

[1]http://en.1d-lab.eu

SEMANTICS 2016: Posters and Demos Track
September 13-14, 2016, Leipzig, Germany

pages, blogs, keywords, attributes and properties for music, and tries to match them to the user's profile [6]. For example Pandora[2], a radio streaming platform, uses as features tags set by experts to recommend music. Since human tagging is a time-consuming task there are methods to retrieve information automatically. [11] uses signal processing on music and inferred high-level semantic descriptors (timber, rhythm, tempo). Other meta-data provided with a track like the music genre can be used. Linked Open Data and Encyclopedic Knowledge Source can also help learning more accurate items features. On the artist's level, [8] propose dbrec, a recommender system based on DBpedia. DBpedia is also used in the movie recommendation field [7]. They use the genres, but also information like the actors, directors, writers, to create a vector for movies and compute similarities with a Semantic version of the classical Vector Space Model (sVSM) that they created. These work require detailed information on each artist or movie that can be recommended. Since we are working with a dataset of independent artists we don't find machine readable data for most of them in the Linked Data Cloud. We therefore choose to use the music genre that is provided with our data to make recommendations. We have seen in [2] that there is no big difference between a content-based semantic distance and a simple genre-based baseline.

We therefore concentrated on the problem of using semantic data to find the semantic relatedness of music genres. There are different works that propose semantic relatedness measures over ontologies [4, 9]. The first proposes a semantic relatedness based on the top-k path connecting two resources. First the top-k paths are found. Then the relatedness is computed based on the properties appearing in these paths. The second is based on the number of different types of connection that exits between two resources x and y. It considers direct connections between x and y and connections where a third resource z is connected to x and y with the same relation.

Here we propose a new semantic relatedness measure. As we have shown in [3] our method has the advantage of considering all relations and classes of the ontology. Also, notice that [9] considers distance between artist in DBpedia, while we are considering music genres, because as previously said, lots of independent artists are not described on the Web nor in the Semantic Web.

3. VECTOR REPRESENTATION AND SEMANTIC RELATEDNESS

In this section we describe how we compute the semantic relatedness for two resources of an ontology. We first associate to each resource r a vector representation that captures the semantic of r. Then, we use this representation to compute the semantic relatedness of two resources.

We interpret the ontology as a graph G. We transverse the graph starting from r up to a depth of k in an directed or undirected way. Let $R_{r,1}$ be the set of resources whose distance from r is 1, $R_{r,2}$ the set of resources whose distance from r is 2 and so on. We construct a vector representation of r which we denote by v_r as follows. v_r lives in a vector-space that has as many dimensions as the number of resources of the ontology. The i-th component is defined as:

$$(v_r)_i = \begin{cases} 0 \text{ if } i \notin R_{r,1} \cup ... \cup R_{r,k} \\ 1 \text{ if } i \in R_{r,1} \\ \vdots \\ \frac{1}{k} \text{ if } i \in R_{r,k} \end{cases}$$

i.e. it contains zero-entries for all resources, except a 1 for the resources in $R_{r,1}$, a 1/2 for the resource $R_{r,2}$ and so on.

To compute the semantic relatedness between two resources r and s we compute the cosine similarity between the two corresponding vectors v_r and v_s:

$$< v_r, v_s >= \frac{\sum_i (v_r)_i \cdot (v_s)_i}{||v_r|| \cdot ||v_s||}$$

This will return a number between 0 and 1. There are two possible interpretations for this similarity measure. The first is that the similarity of the resources is based on the angle between the corresponding vector representations. The second is to interpret it as the overlap of some sub-graphs of G. The vector of r contains non-zero entries for all resources that are at a distance of maximum k from r. Let G_r be the sub-graph containing these resources. Analogously for the vector of s. This means that the sum in the cosine similarity is non-zero for all resources that are at a distance of k from both r and s, i.e. we compute how big is the overlap between the sub-graphs G_r and G_s.

4. SEMANTIC RELATEDNESS OF MUSIC GENRES

As an application of our similarity measure we have computed the similarity between music genres. As an ontology we considered the English DBpedia[3] ontology. We created a list of music genre. Many of them could be identified using the *dbo:MusicGenre* class. In a next step we computed the vector representation of all music genre. We constructed the vectors as described above by setting $k = 2$ and by considering the ontology graph as an undirected graph. Note that for a music genre r the corresponding vector-representation contains non-zero entries for example for: instruments that are characteristic for the genre, the region where it comes from and it's stylistic origin. Moreover since we consider the graph as undirected it contains also bands that play this genre, albums and tracks that are of this genre and many more. We do not restrict the search to a particular properties but we consider them all.

All the computations where done using the work presented in [3]. It shows how to transverse efficiently an ontology using sparse matrix multiplications and it was used to construct the vector representations presented above.

We used the vector representation to compute the pairwise semantic relatedness of the music genre. The result is a square dense matrix with one row and one column for each music genre. The i-th row j-th column contains the semantic similarity between the music genre i and j, i.e. a double between 0 and 1. Since it is difficult to judge the quality of the result in this format we decided to visualize them. We do not consider all connections, but using a threshold, only the connections with a high degree of similarity. Otherwise it would be difficult to visualize all the data, since we have 132

[2]pandora.com

[3]http://dbpedia.org

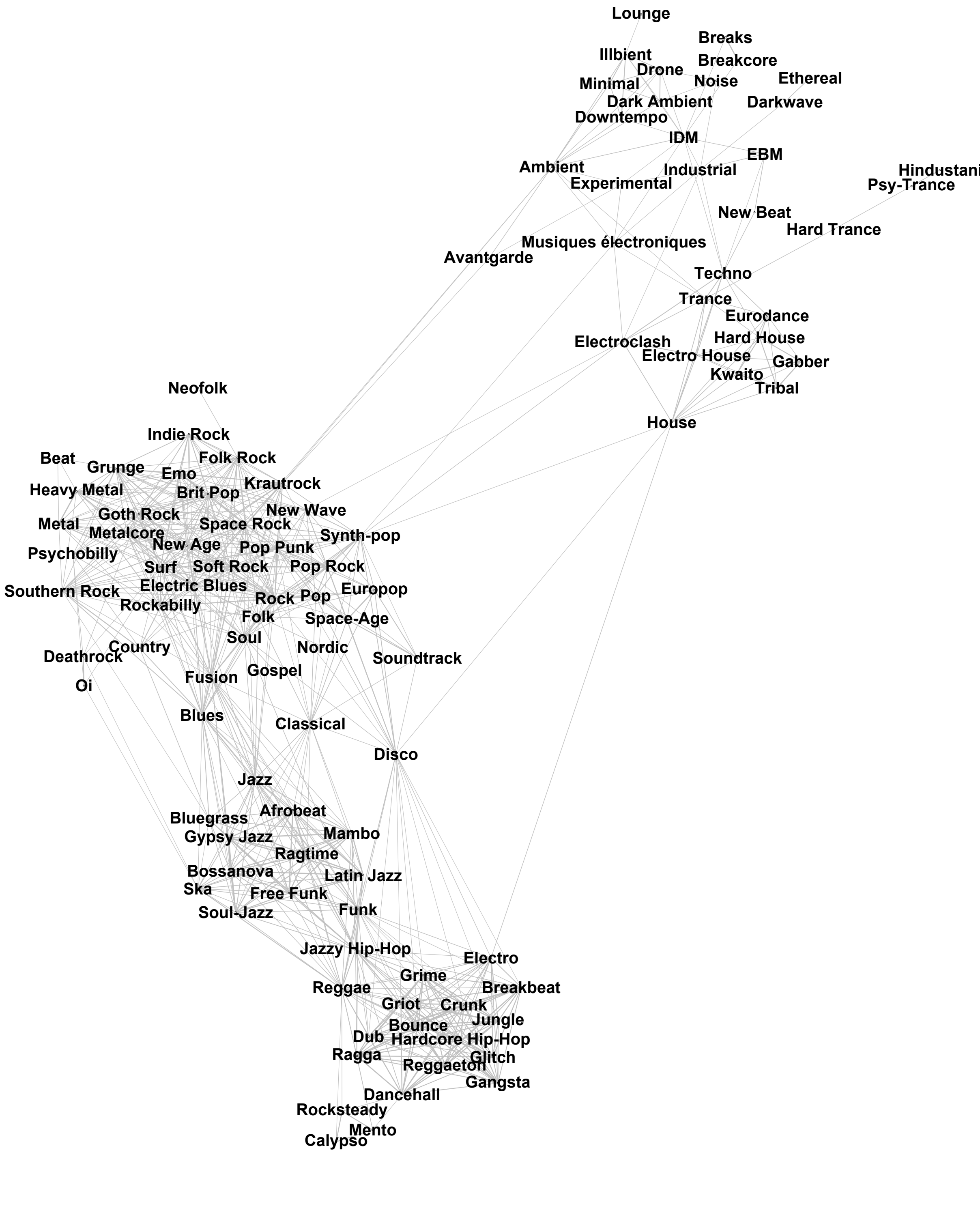

Figure 1: Graph of the music genre constructed using the corresponding semantic relatedness and visualized using the Force Atlas 2 algorithm of Gephi

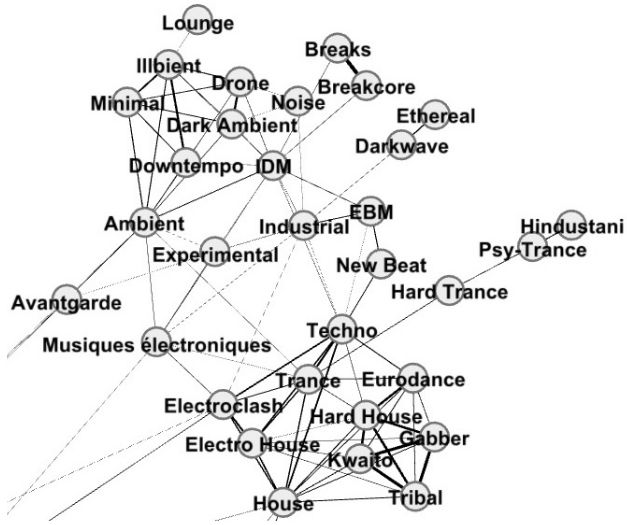

Figure 2: Graph of electronic music genre constructed using the corresponding semantic relatedness and visualized using the Force Atlas 2 algorithm of Gephi

nodes and 37044 edges, i.e. nearly all nodes are connected between each other.

The data is displayed using the Force Atlas 2 [5] algorithm provided by Gephi [1]. The main idea behind the visualizing algorithm is that all nodes repulse each other while two connected nodes attract each other.

The entire graph is showed in figure 1 while a subgraph including electronic music genres is showed in figure 2. Both show that we were able to capture the semantic relatedness between the music genres.

5. FUTURE WORKS

We are working on a music recommendation system for independent artists using the numbers that we presented here. Moreover we would like to compare our similarity measure to existing ones and see how it performs. Another interesting direction is to make the semantic relatedness more flexible. The semantic relatedness between music genre is not absolute. A user could be more interested in music genre that come from the same geographical area, music genre that arose in the same period of time or that are characterised by the same instruments. By weighting accordingly different relations it would be possible to reflect these interests and give accordingly recommendations to the users.

6. ACKNOWLEDGMENTS

This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement No 642795.

7. REFERENCES

[1] M. Bastian, S. Heymann, M. Jacomy, et al. Gephi: an open source software for exploring and manipulating networks. *ICWSM*, 8:361–362, 2009.

[2] D. Bogdanov, M. Haro, F. Fuhrmann, E. Gómez, and P. Herrera. Content-based music recommendation based on user preference examples. *Copyright Information*, page 33, 2010.

[3] D. Diefenbach, R. Usbeck, K. D. Singh, and P. Maret. A scalable approach for computing semantic relatedness using semantic web data. In *Proceedings of the 6th International Conference on Web Intelligence, Mining and Semantics*, WIMS '16, pages 20:1–20:9, New York, NY, USA, 2016. ACM.

[4] I. Hulpuş, N. Prangnawarat, and C. Hayes. Path-based semantic relatedness on linked data and its use to word and entity disambiguation. In *International Semantic Web Conference*, pages 442–457. Springer, 2015.

[5] M. Jacomy, T. Venturini, S. Heymann, and M. Bastian. Forceatlas2, a continuous graph layout algorithm for handy network visualization designed for the gephi software. *PloS one*, 9(6), 2014.

[6] P. Lops, M. De Gemmis, and G. Semeraro. Content-based recommender systems: State of the art and trends. In *Recommender systems handbook*, pages 73–105. Springer, 2011.

[7] R. Mirizzi, T. Di Noia, A. Ragone, V. C. Ostuni, and E. Di Sciascio. Movie recommendation with dbpedia. In *IIR*, pages 101–112. Citeseer, 2012.

[8] A. Passant. dbrec music recommendations using dbpedia. In *International Semantic Web Conference*, pages 209–224. Springer, 2010.

[9] A. Passant. Measuring semantic distance on linking data and using it for resources recommendations. In *AAAI spring symposium: linked data meets artificial intelligence*, volume 77, page 123, 2010.

[10] P. Resnick and H. R. Varian. Recommender systems. *Commun. ACM*, 40(3):56–58, Mar. 1997.

[11] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet. Semantic annotation and retrieval of music and sound effects. *Audio, Speech, and Language Processing, IEEE Transactions on*, 16(2):467–476, 2008.