Co-creation with AWWZ: "AI b2b DJ"

Ioannis Tsiamas
ioannis.tsiamas@upc.edu

1 Introduction

The objective of this co-creation project was to create an Artificially Intelligent (AI) DJ system that can play a b2b DJ along a human DJ, AWWZ[1], at the AI and Music S+T+ARTS Festival[2]. The AI DJ system is basing its track selection on the sentiment, perception or feelings that musical tracks create on people. We used the public comments from YouTube as a proxy for that feeling and encoded them through a pre-trained Language Model to" understand" the language. The AI DJ system plays the role of a human DJ, by selecting tracks that are similar to the feeling of the recent history of tracks in the DJ set. A central point and innovation of this project is that the data modality the AI DJ system is using, is not music, but text, which makes it purely language-based and sound-agnostic.

2 Co-creation Process

A b2b (back-to-back) DJ set is a collaborative DJ set, where two (or more) DJs play tracks, alternating the selection between them. This creates a dynamic setting, where each DJ has to base their decisions not only on their track selection but also on the selection of the others. Due to this dynamic nature, we believe it was a perfect opportunity to develop an AI system that can take the place of one of the human DJs, and allow human and machine to co-operate and create a 40-minute musical experience for the attendees of the festival.

The co-creation process started with meetings between the artist, AWWZ, the researchers from UPC and representatives from the festival, to examine the different possibilities for the project, and establish the time frame and necessary hardware and software requirements. After the first initial meetings, decisions were made on the size of the track-list, the AI methods that would be used to model the tracks and obtain track representations, how will the AI DJ system use these representations during the b2b DJ set, and which software and libraries will be employed to communicate information between the AI and the human DJ during the live. Thus, the team started working on three tasks: (a) Data scraping and cleaning, (b) AI modeling, and (c) AI-human DJ communication. The team that was working on these three tasks, was comprised of two PhD researchers from UPC, Ioannis Tsiamas and Casimiro P. Carrino, master's student, Mireia De Gracia, and UPC professor, Marta R. Costa-jussà. At the final part of the co-creation, Dimas A. Martinez was added to the team to assist with the software engineering part of the project.

The "Data scraping and cleaning" task was the first to be completed, where the artist provided a track-list of 900 tracks, which were identified on YouTube and their comments were scrapped, using automated scripts. Subsequently, the "AI modeling" part started taking place. The comments of each track were modeled by a pre-trained language model to obtain meaningful representations in a semantic space, where tracks with similar comments are close and tracks with dissimilar are far. The AI DJ system is using this semantic space to make decisions on track selections during the DJ set. The team, together with AWWZ evaluated several semantic spaces, before deciding on the hyperparameters of the AI DJ system, that would lead to the optimal result. At the same time tests were carried out for the "AI-human DJ communication", and a communication pipeline was established, where the AI DJ system uses Ableton Live[3], the human DJ uses Traktor[4], and the two DJ software are linked together. Furthermore, PyLive[5] was used to control Ableton Live with python.

Finally, after the completion of the three tasks in late September, presential meetings were held at a space provided by the artist, to carry out rehearsals for the b2b DJ set, identify and correct unexpected errors in the systems, and do last minute calibration of the AI DJ system. During these meetings, we established some additional restrictions to the AI DJ system, which aimed at artist diversity (avoiding same artist repetition) and bpm (beats-per-minuted) consistency (choosing tracks close to the bpms of the previous one). We also

[1]https://soundcloud.com/awwz
[2]https://aimusicfestival.eu/
[3]https://www.ableton.com/en/live/
[4]https://www.native-instruments.com/en/products/traktor/dj-software/traktor-pro-3/
[5]https://github.com/ideoforms/pylive

accounted for the possibility of the artist, to play tracks outside of the known to the AI track-list by allowing the AI DJ system to base its decisions on the recent history of tracks and not only on the very last one, as was originally intended.

Important to the success of the co-creation was the frequent and constructive collaborations between the team members, as well as the input of the artist, AWWZ, especially on helping with the musical and artistic aspects of the project. All the team members and the artist were delighted with the final outcome during the festival, which was also well received from the audience and the organizers. Given the good reception of the AI b2b DJ set, as well as the fact that all team members enjoyed and learned a lot throughout the co-creation process, we are looking forward to contributing to more co-creation projects in the future, and experimenting further with the integration of Artificial Intelligence methods in the musical and artistic landscape.

3 Technical Approach

Given the initial track-list of approximately 900 tracks (track name, artists, average bpm, duration), we query each track name along with track artists using the youtubesearchpython[6] library and retrieve the top-10 results. We have to make sure that we identify the correct corresponding youtube video for each track thus we select the result that satisfies a set of requirements. These are (a) the inclusion of at least one artist in the title or description, (b) a maximum of 20 seconds of difference between youtube time and track-list time of the track, (c) a minimum of 10 comments. Then we used youtube-comment-downloader[7] library to scrape the top-5000 comments (text likes) for each one of them. As internet text data are usually very noisy, we employed several data cleaning steps to process the comments and ensure that our data are of good quality, for the language model to model them properly. Thus, we handled non-utf8 and Html characters, normalized spacing, punctuation and emoticons/emojis, removed comments that were either too short or too long and finally identified and discarded potential spam comments. The data scraping and cleaning process resulted in a dataset of 387 tracks accompanied by a total of 350,000 comments.

We use a pre-trained Language Model to map the text of each comment to a semantic vector space, which is usually referred to as an embedding. The pre-trained Language Model has been trained for multilingual sentence similarity with millions of examples, making it suitable for producing meaningful representations of comments. This means that similar comments are mapped close to each other and dissimilar ones are mapped far from each other, as measured with their cosine similarity. We specifically use the Sentence-Transformers [4] library and the checkpoint of "paraphrase-multilingual-mpnet-base-v2", which uses XLM-RoBERTa-base [1] as a teacher, a transformer encoder [5] with 12 layers and a total of 270 million parameters. The checkpoint of the student model was trained with XLM-RoBERTa in a Siamese network setting [3], where two identical copies of the teacher model encoder a pair of sentences and the cosine-similarity is used as a loss function. Apart from the very high performance of this model on the sentence similarity task, the decision of this specific model was also motivated by the fact that the model is multilingual and can also handle emojis, which are both important aspects of the dataset we created. For reference, we used FastText [2] and identified 34% of the comments as non-English. Also, 20% of the comments contain emoticons/emojis.

After obtaining the comment embeddings of each track, we aggregate them using a like-assisted weighted-mean function into track embeddings. This aggregation function was found to perform much better than a simple mean or max pooling. We assume that the likes of each comment are useful to down-weight noisy and spam comments that we were not able to filter out during data cleaning, and thus the obtained track embeddings are more meaningful and capture better the semantics of the tracks.

$$y_{i,j} = SentenceTransformer(x_{i,j}^{text}) \tag{1}$$

$$Y_i = \frac{\sum_j y_{i,j} \cdot x_{i,j}^{likes}}{\sum_j x_{i,j}^{likes}} \tag{2}$$

$$Sim_{a,b} = \frac{Y_a Y_b}{|Y_a| \cdot |Y_b|} \tag{3}$$

Where $x_{i,j}^{text}$ and $x_{i,j}^{likes} \in N$ is the text and likes of the j-th comment for the i-th track, $y_{i,j} \in R^d$ is its comment embedding, $Y_i \in R^d$ is the embedding of the i-th track, and $Sim_{a,b} \in (0, 1)$ is the cosine similarity between tracks a and b.

[6]https://github.com/alexmercerind/youtube-search-python

[7]https://github.com/egbertbouman/youtube-comment-downloader

The track embeddings lie in the same embedding space as the comments and can be viewed as a representation of the "average" comment in each track. Since the properties of the space were preserved, similar tracks are grouped close together and dissimilar ones are far. We can quantify their distance using cosine similarity, the metric with which the model was trained.

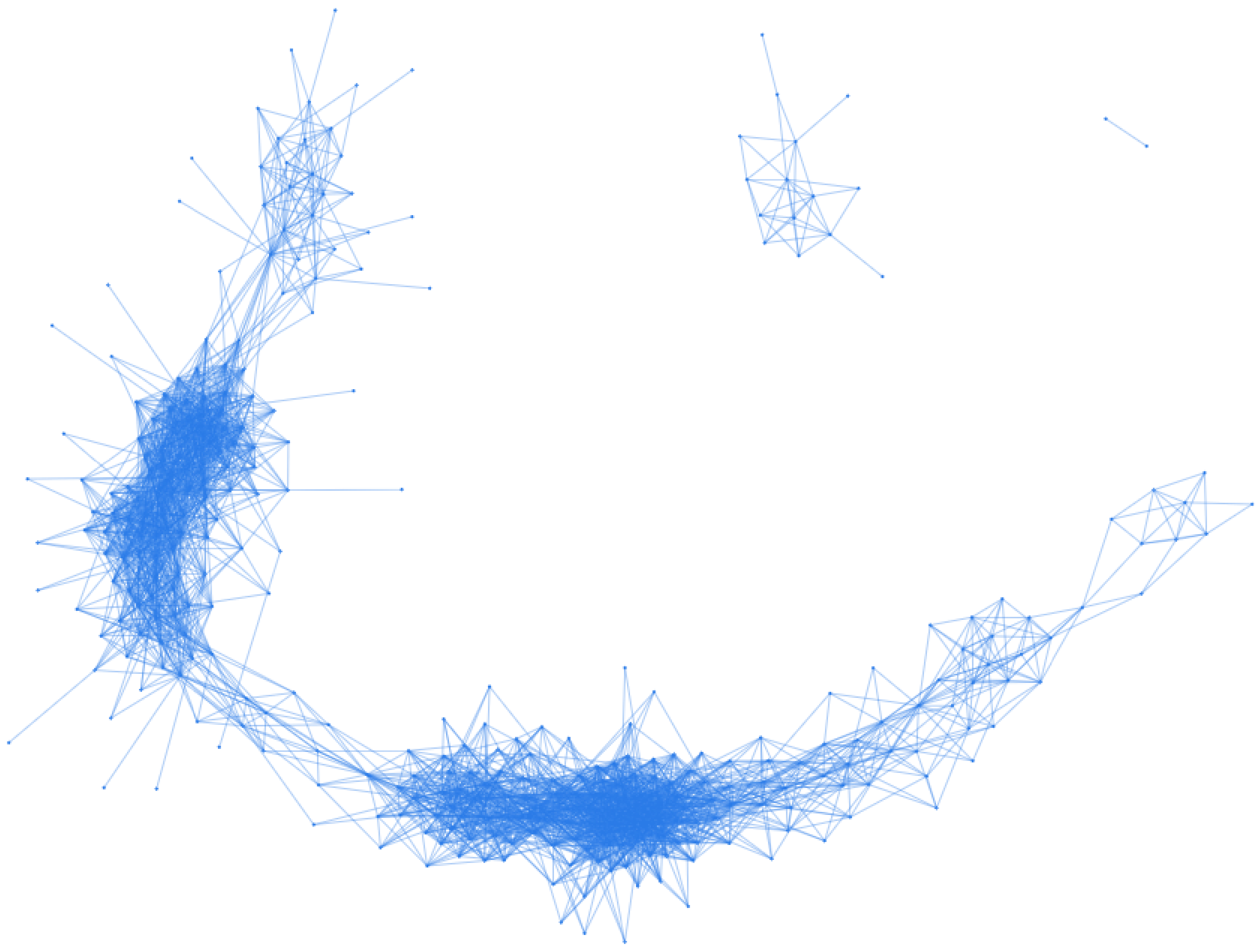

Figure 1: Track similarity graph extracted from the similarity matrix. Each edge is a track and each node the similarity between two tracks. For ease of representation only nodes with similarity higher than 0.75 are plotted.

We computed the similarities between all the tracks in our track-list using the above process and we thus obtained an $N \times N$ similarity matrix (Fig. 1) that can be used by the AI DJ system during the b2b DJ set. At any step during the b2b DJ set, when the AI DJ system is requested to play a track, given a history of n already played tracks (by both DJs) the following procedure (Fig. 2) is taking place:

1. Retrieve the similarity vectors that correspond to the $m = 5$ most recent tracks
2. Aggregate the m similarity vectors into one similarity vector with a weighted average. The most recent track is assigned a weight of 50%, which decay exponential for the rest of the tracks. This is an N-dimensional vector, where its i-th element is the similarity score of the i-th track in our track-list, with the recent history of tracks.
3. Zero-out the similarity scores of tracks according that are from artists already played during the set, or have more than 5 bpms of difference with the last track that was played.
4. Normalize the similarity vector to obtain a probability distribution over the track-list
5. Sample the next track using greedy sampling

Qualitative and quantitative evaluations were performed to assess the quality of the embedding space and tune hyperparameters such as the aggregation function and the number of comments to be used for each track. The qualitative evaluation was carried out by the artist, to which we provided DJ sets, that were generated from

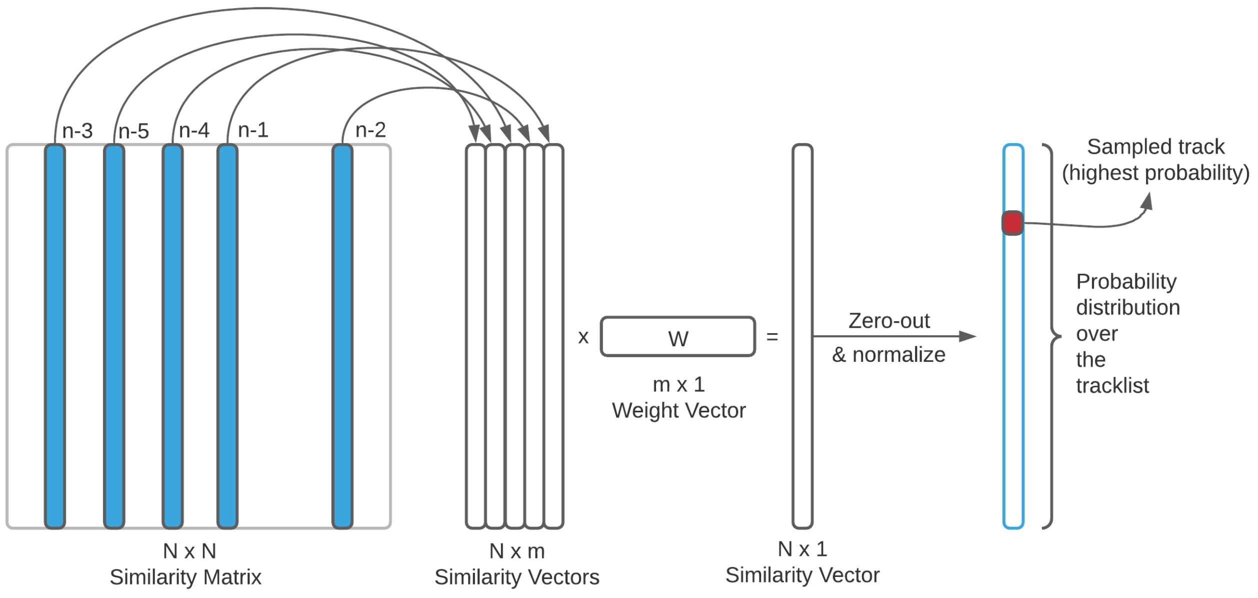

Figure 2: Visual representation of a track selection step from the AI DJ system

the AI DJ system, by starting with a random first track. The quantitative evaluation was based on real track transitions that were extracted from previous DJ sets of the artist. Using these are ground truth transitions, we evaluated several iterations of the AI DJ system by calculating the probability that the model assigned to them.

After the track selection process from the AI DJ system is completed, the information is communicated with PyLive to Ableton Live, which mixes the new track on top of the one played by the artist, using its auto-mixing function, which adjusts the bpms of the new track to ensure an acoustically smooth transition.

The code that was used for data scraping and cleaning, obtaining the track embeddings, and the AI DJ system application are available in a public repository[8].

Possible directions for further research is the experimentation with larger track-lists containing diverse styles of music. Although we used a pre-trained language model for this project, we believe that we can obtain better comment/track representations by fine-tuning the model on a large scrapped corpus of YouTube comments to reduce the domain mismatch with the original training data. Moreover, combining the sentence similarity model with an emotion classifier will add extra information that may better extract the “mood” of the comments. Lastly, the AI DJ system can benefit from more sophisticated decoding algorithms, such as beam-search, instead of using greedy sampling to select the next track.

[8]`https://github.com/mt-upc/upc-sonar-2021`

References

[1] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale, 2020.

[2] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.

[3] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.

[4] Nils Reimers and Iryna Gurevych. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2020.

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.