S. Venkatesh, D. Moffat and E. R. Miranda, "Word Embeddings for Automatic Equalization in Audio Mixing"
J. Audio Eng. Soc., vol. 70, no. 9, pp. 753–763, (2022 September).
DOI: https://doi.org/10.17743/jaes.2022.0047

Word Embeddings for Automatic Equalization in Audio Mixing

SATVIK VENKATESH,[1]* DAVID MOFFAT,[2] *AES Member* **AND EDUARDO RECK MIRANDA[1]**
(satvik.venkatesh@plymouth.ac.uk) (dmof@pml.ac.uk) (eduardo.miranda@plymouth.ac.uk)

[1]*Interdisciplinary Centre for Computer Music Research, University of Plymouth, Plymouth, UK*
[2]*Plymouth Marine Laboratory, Plymouth, UK*

In recent years, machine learning has been widely adopted to automate the audio mixing process. Automatic mixing systems have been applied to various audio effects such as gain-adjustment, equalization, and reverberation. These systems can be controlled through visual interfaces, audio examples being provided, usage of knobs, and semantic descriptors. Using semantic descriptors or textual information to control these systems is an effective way for artists to communicate their creative goals. In this paper, the novel idea of using word embeddings to represent semantic descriptors is explored. Word embeddings are generally obtained by training neural networks on large corpora of written text. These embeddings serve as the input layer of the neural network to create a translation from words to equalizer (EQ) settings. Using this technique, the machine learning model can also generate EQ settings for semantic descriptors that it has not seen before. The EQ settings of humans are compared with the predictions of the neural network to evaluate the quality of predictions. The results showed that the embedding layer enables the neural network to understand semantic descriptors. It was observed that the models with embedding layers perform better than those without embedding layers but still not as well as human labels.

0 INTRODUCTION

The process of audio production involves multiple tasks such as balancing sound levels and applying audio effects. An audio effect can be defined as a function that transforms sound based on a set of controlled parameters [1]. Audio production is needed in various domains such as making albums, films, and theater works, to name a few. It is generally carried out by a mixing engineer who understands the goals of their client. The mixing engineer blends multiple tracks together by modifying acoustic properties such as dynamics and timbre [2]. A vast body of research has been exploring how this process can be automated through the use of *intelligent tools* [3–6]. Traditional Artificial Intelligence (AI) approaches such as expert systems have been adopted to create autonomous mixing tools [3]. These systems are knowledge-engineered and adopt a set of rules for mixing depending on the scenario. However, recent research has grown toward using Machine Learning and Deep Learning for automatic mixing. On one hand, some studies have focused on specific areas, such as gain balancing [7] and reverberation [8]. On the other hand, some have explored building autonomous systems in which the entire mixing process is carried out without human intervention [2, 9].

An equalizer (EQ) is an audio effect created by cascading multiple filters in series [10]. Timbral adjectives often have a correlation with the parameter setting for the equalizer. Some examples include *add air*, *make it warmer*, and *make it less muddy* [11]. Kulka [12] associated adjectives such as *warmth*, *honk*, *crunch*, and *sibilance* with frequencies of 125, 500, 2,000, and 8,000 Hz, respectively. For example, according to the Kulka rule, if the mix sounds honky, cut the region around 500 Hz.

When clients such as instrumentalists and musical directors work with mixing engineers, they often use semantic descriptors to describe their goals. For example, "make the violin sound warmer" [13]. It is the role of the mixing engineer to understand these descriptors. Popular semantic descriptors such as *warm* and *bright* are easily understood by the mixing engineer [14]. To expand the vocabulary of such descriptors, studies have also tried to create a thesaurus with synonyms and antonyms. For example, significant synonyms of *boom* are *boxy*, *dull*, and *fat* and significant antonyms of *boom* are *air*, *bright*, and

*To whom correspondence should be addressed e-mail: satvik.venkatesh@plymouth.ac.uk

crisp [11]. However, the problem arises when individuals without training in audio production describe their creative goals [15]. They may have ideas that cannot be directly translated into a studio engineer's vocabulary.

To address this issue of non-technical descriptors, Cartwright and Pardo [13] presented a dataset called SocialEQ, which is a web-based project that adopts crowdsourcing to learn a vocabulary of audio descriptors. Because it is crowdsourced, the study focuses on aggregating a vocabulary to enable non-technical individuals to describe their sonic goals. Crowdsourcing was also adopted to build the datasets for other effects like reverberation [16] and dynamic range compression [15].

There is a growing interest in adopting natural language processing (NLP) methodologies to develop semantically controlled audio effects [17–19]. Stables et al. [20] presented a system called Semantic Audio Feature Extraction (SAFE), which focused on extracting semantic descriptions for equalization from a digital audio workstation (DAW). Stasis et al. [21] investigated the idea of mapping the descriptors to a reduced dimensionality space, to enable users to interact with the system in a more intuitive way. Chourdakis et al. [22] explored tagging and retrieval of room impulse responses for reverberation. They adopted word embeddings to assign impulse responses to tags that match their short descriptions.

This paper explores the novel idea of adopting word embeddings to automatically predict EQ settings. A methodology is presented to translate words from a semantic vector space to a vector space representing the parameters of an equalizer. Word embeddings are representations of words that capture lexical semantics in language [23]. An embedding layer is often used as the first layer in a neural network that performs NLP tasks, such as machine translation, caption generation, and automatic speech recognition [24]. Although word embeddings are commonly used to understand natural language, this paper investigates whether they would be of any benefit to descriptors for EQ settings. This approach is adopted to translate words to predict values of a parametric equalizer. This way, the neural network has the ability to understand non-technical words and even descriptors that it has not seen before. This finding is significant because artists without training in audio production can express their creative goals directly to the AI-powered mixing engine. To the authors' knowledge, this is the first study that investigates how EQ settings can be predicted for *unseen* semantic descriptors. It is demonstrated that the neural network is capable of learning a direct translation from the text domain to the EQ domain.

1 METHODOLOGY

1.1 Dataset

The SocialEQ dataset [13], which crowdsources semantic descriptors for EQ settings, is adopted. In the raw format, each sample in the dataset contains a semantic descriptor, language of the descriptor, audio identifier, a consistency rating, and 40 values for EQ parameters. During the data collection, each participant was asked to enter a word in their preferred language. For example, *warm* in English, *claro* in Spanish, or *grave* in Italian. Subsequently, they picked a sound file, which was modified by the EQ plugin. There were three sound files—electric guitar, piano, and drums. Each sound file had a unique audio identifier.

After selecting a descriptive term and audio file, the participant was presented with 40 different modifications of the sound file made by different EQ settings. If the user selected *warm*, they were asked to rate *how warm that sound is*. Out of the 40 modifications, there are 15 repetitions to test for consistency. Consistency score was calculated using Pearson correlation between the ratings of the test and repeated examples. The system processes the ratings of the user and develops a relative boost/cut for 40 different frequency bands. Refer to Cartwright and Pardo [13] for more details on the dataset.

The dataset has 1,595 samples in it. For simplicity, only descriptors in English were considered. The number of examples in English was 918. It is important to note that the dataset contained examples with different EQ parameter settings for the same word. Thus, the number of unique descriptors in English was 388.

1.2 Train-Test Split

An important hypothesis the authors wanted to test in this paper is that a word embedding layer helps a model predict EQ parameter settings for semantic descriptors it has not seen before. Therefore, words in the test set should not appear in the training set. A four-fold cross-validation setup [25] was adopted, and the strategy is explained below.

A list of semantic descriptors that are common in the audio mixing literature was aggregated. These were labeled high-quality (HQ) words. In order to avoid bias and objectively choose these words, those that were already listed in Table 4.8 in [11] were selected. Additionally, semantic descriptors that fell under the hierarchical ontology presented by Pearce et al. [26] were included. The list of HQ words is presented in bold in Table 1. There are 32 HQ words present in the SocialEQ dataset.

A list of words that were highly rated (HR) was also aggregated. HR words need not be semantically meaningful but do need to have a high consistency score in the dataset. Words that have a consistency score greater than 0.7 were selected as HR words. Because these words have a high consistency score, the user strongly associated the semantic word with a particular EQ setting. Words in Table 1 that are not formatted as bold text are HR words. Totally, 86 HR words were present in the SocialEQ dataset.

Each test fold contained nine HQ and 22 HR words. It was ensured that every HQ and HR word was tested at least once. In the last test fold, there may be a few repetitions of words from the first test fold. There was no overlap between the training and test sets. The test set only contained words that were not present in the training set. Note that the network for each fold is trained as a separate experiment. In other words, the network is totally trained four times and tested

Table 1. Four cross-validation folds from the dataset. The test words from each fold are presented in the table. For each fold, the training set consists of words that are not in the test set.

Fold 1	Fold 2	Fold 3	Fold 4
smooth, muffled, crisp, punch, clean, brittle, muddy, soothing, clear, brassy, caring, mellow, throbbing, cooing, fluffy, good, excited, squeaking, punchy, funky, whispered, disgusting, beautiful, reserved, serene, thumpy, pleasurable, whispering, gentle, energetic, peace	**crunchy, woody, flat, metallic, dull, tinny, cold, booming, deep,** energizing, heart-warming, edgy, heavy, edge, strong, enchanting, cheerful, plodding, quiet, radiant, biting, brass, pleasing, light, taco, gruff, exciting, love, heat, techno, solemn	**sweet, warm, airy, full, boxy, bright, boom, fat, shrill,** calm, velvety, hard, rich, noisy, down, rumble, sloppy, relaxing, peaceful, romantic, low, hot, thunderous, frigid, happy, poor, cool, tense, jagged, forceful, aggressive	**sharp, big, dark, hollow, harsh, smooth, muffled, crisp, punch,** mournful, clarity, genius, bold, twangy, soft, splash, slow, wistful, brash, fancy, cute, rousing, loud, breezy, large, passionate, baseball, huge, icy, brassy, caring

four times on different folds, and the average performance is reported.

As mentioned earlier, each word can have multiple EQ settings. Each setting is a separate example and can have different consistency scores. In the test set, only examples that had a consistency score greater than 0.7 were included. In the training set, no words were excluded based on the consistency score.

1.3 Word Embeddings

A vocabulary consists of all the possible words that the neural network can understand. Generally, a word is converted into a one-hot encoded vector before passing into the neural network. For instance, in the SocialEQ dataset, there are 388 unique words, which means that the size of the vocabulary is 388. Therefore, the dimensions of the one-hot encoded vector are 1×388. Each position within the vector is assigned to a unique word. Thus, the respective position of the word is labeled as 1, and the remaining elements are 0. However, it is important to note that the Euclidean distance between any pair of words is equal. Because each word is equidistant from each other, the neural network is not capable of handling words that are not present in the training set. For example, consider the semantic descriptor *bright*, and assume that it is present in the training set. Also assume that *clear* and *boom* are words in the test set. According to Stasis [11], *clear* is a synonym of *bright*, and *boom* is an antonym of *bright*. Thus, similar EQ settings are expected for *clear* and *bright*, but considerably different EQ settings are expected for *boom* and *bright*. However, the neural network cannot perceive this understanding unless it has seen all three words because each word is equidistant from each other. Furthermore, this issue becomes exaggerated if a non-technical user is utilizing a semantic descriptor that is not common in the audio mixing literature.

A word-embedding layer converts a one-hot encoded representation into a vector space of reduced dimensionality. Large vocabularies with millions of words can be reduced to a 300-dimensional vector representation [27]. The distances between words in the embedding space are governed by some form of semantic correlation. Examples include synonyms or two words frequently occurring together. There are different algorithms to train word-embedding models. Some of them include Word2Vec [28], GloVe [27], ConceptNet [29], and Dict2Vec [30]. Each of these algorithms presents unique methods to train on large corpora of text, such as Wikipedia. Effectively, they try to learn semantic relationships between words and represent them through an embedding vector.

This study investigated four different embedding models—GloVe-6B, GloVe-840B, Tok2Vec, and Dict2Vec. GloVe is an unsupervised learning algorithm developed to obtain vector representations for words [27]. GloVe-6B refers to the model that was trained on Wikipedia 2014 and Gigaword 5. It includes 6 billion tokens and a vocabulary size of 400,000. Moreover, GloVe-840B uses 840 billion tokens and a vocabulary size of 2.2 million. It trains on the World Wide Web using Common Crawl, which is a larger corpus of text. Tok2Vec is a word-embedding model provided by a company called spaCy [31]. The entire details regarding its implementation were not found, but the model is publicly available and free to use.

It is important to note that word-embeddings are used for NLP tasks, which are designed to accept sentences. In this application, the authors are considering only one word, which is the semantic descriptor. Because GloVe and Tok2Vec also focus on the ordering of words in sentences, the authors thought it was a good idea to consider another embedding model called Dict2Vec [30]. Dict2Vec is an embedding model that uses lexical dictionaries. It builds new word pairs from dictionary entries so that semantically-related words are closer to each other in the embedding space [30]. Similar to GloVe-6B, it was trained on the Wikipedia corpus.

1.4 Machine Learning Architecture

1.4.1 Word-Embedding Layer

Four different pre-trained word-embedding models were evaluated in the study—GloVe-6B, GloVe-840B, Tok2Vec, and Dict2Vec. All the models represent words with 300-dimensional semantic vectors. This is convenient because the same neural network architecture can be adopted to compare different embeddings. Initially, a word is converted into a one-hot encoded representation. Subsequently, an embedding matrix converts this one-hot encoded representation into a 300-dimensional semantic vector. Then,

Table 2. The neural network architecture.

Layer type	Units	Activation	Output shape
Embedding	···	···	300
Dense	300	ReLu	300
Dense	200	ReLu	200
Dense	100	ReLu	100
Dense	80	ReLu	80
Dense	60	ReLu	60
Dense	40	Sigmoid	40

ReLu, rectified linear activation unit.

this vector is connected to hidden layers in the network. Note that the weights of the embedding matrix are frozen and the layer is not trainable. The authors did not consider setting this to trainable because of the limited data they have.

1.4.2 Hidden Layers

The neural network aims to translate a representation of word embeddings to a prediction of equalizer parameters. Therefore, this network needs to be deep enough to learn the translation between two domains. Deeper networks apply the non-linear activation more times on the input and therefore have the advantage of learning more complex translations. However, it is important to note that the dataset is relatively small for this task.

All the layers in the neural network were fully connected layers. Table 2 shows an overview of the architecture. After the embedding layer, there was a series of fully connected layers. The number of hidden units in these layers were 300, 200, 100, 80, and 60, respectively. Finally, it was connected to an output layer with 40 units. Excluding the final layer, all the hidden layers were fitted with rectified linear unit activations and a dropout of 0.1. The output layer is explained in Sec. 1.4.4. The code and trained models associated with this study can be found in this GitHub repository (https://github.com/satvik-venkatesh/word-eq).

1.4.3 Normalization

Traditional min-max normalization by calculating the maximum and minimum in the training set was not appropriate for this dataset. This is because if there exist any outliers among the values in the test set, specific features may get magnified or diminished. Furthermore, because values for 40 EQ bands are being predicted, this issue becomes more crucial. Therefore, the minimum and maximum value for each EQ parameter were fixed to –4 and +4 dB, respectively. In other words, the highest cut/boost within each EQ band was 4 dB. The values were linearly normalized to the range of 0 to 1. Hence, –4 dB would correspond to 0, and +4 dB would correspond to 1 in the output layer.

1.4.4 Output Layer and Loss Function

The output layer of the network contained 40 neurons, with each of them predicting a value for one EQ band. Because the data were normalized within the range of 0 to 1, sigmoid activation functions were used for the output neurons. The loss function was the mean absolute error, which is commonly used by studies for regression tasks. All EQ bands were given equal importance when averaging the error for the loss function. In future work, it would be interesting to weigh the EQ bands based on perceptual frequency band weights. However, that is beyond the scope of this study.

The network was trained using stochastic gradient descent with an initial learning rate of 0.1. The learning rate was scaled by 0.96 after every 10,000 weight updates.

Table 3. The error calculated across four folds. The smallest error is indicated in bold.

Word embedding	Error
Tok2Vec	$\mathbf{0.760 \pm 0.055}$
Glove-840	0.770 ± 0.032
Dict2Vec	0.792 ± 0.058
Glove-6B	0.798 ± 0.046
No Embedding	0.836 ± 0.016

2 RESULTS

2.1 Error

Table 3 shows the mean absolute error for different embedding models calculated across four test folds. As can be seen, Tok2Vec obtains the lowest error of 0.76, followed by GloVe-840 with an error of 0.77. GloVe-840 obtains an error lower than GloVe-6B, which conveys that it benefited from training on a larger corpus. Dict2Vec and GloVe-6B were trained on similar dataset sizes, and the former obtained a better error. This suggests that the performance of Dict2Vec can be improved with training on a larger corpus of text.

The No Embedding model in Table 3 means that no word-embedding layer was used in the neural network. This can be considered to be the baseline system. Because this is the first study that investigates a translation from *unseen* semantic descriptors to EQ settings, there are no state-of-the-art approaches for comparison. The input of the network was a direct one-hot encoded representation. All the neural networks with word embeddings performed better than the model without word embeddings. However, the difference was not huge. The best model was Tok2Vec with an error of 0.76 vs. No Embedding with an error of 0.836. This is possibly due to two reasons. Firstly, error may not be the best metric for this task. For example, the semantic word *warm* may have a boost of 1.2 dB at 260 Hz. But the neural network may predict a boost at the adjacent EQ band, such as 317 Hz. Although the the error in this case is high, the EQ effect applied to the audio may still be semantically meaningful. Secondly, the test set contains many semantic descriptors that occur only once. These examples may be highly subjective to one individual, despite having a high consistency score. Therefore, in the next subsection, the top two performing models are evaluated using Partial Curve Mapping (PCM) [32], which is a method to quantify the similarity between two curves. For instance, this technique

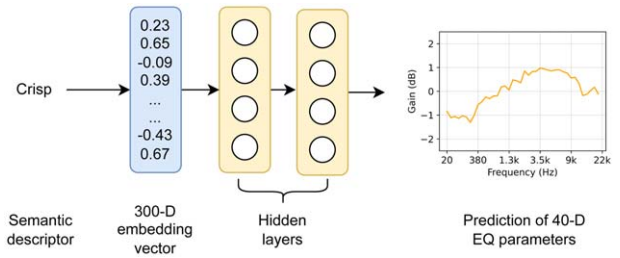

Fig. 1. A schematic diagram of how the network learns a translation from semantic descriptors to equalizer (EQ) parameters.

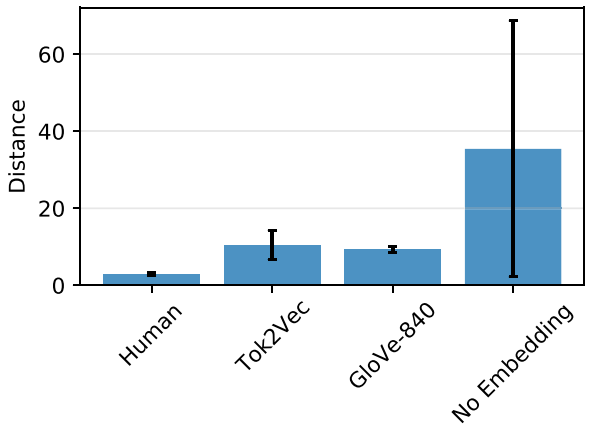

Fig. 2. Distances obtained by different models calculated by using Partial Curve Mapping (PCM). An ideal algorithm would have a distance of zero.

is generally adopted to analyze similarities between hysteresis curves pertaining to a magnetic field. Although this technique may not be ideal for this task, it would give a better understanding of this model's performance compared to mean absolute error.

2.2 PCM

In this section, the models are evaluated using PCM. PCM was implemented using this Python package [33]. The models are also compared to human labels. As mentioned earlier, each semantic descriptor had multiple EQ settings in the dataset. To calculate the error in human labels, the mean of the different EQ settings was considered the ground truth. However, words that occur only once in the dataset would not have an error associated with it. These words would artificially reduce the average error. Hence, only words that occur at least twice in the dataset were included. Fig. 2 shows the distances for different models. An ideal algorithm would obtain a distance of zero. Human labels obtain the smallest distance of 2.9, which is an expected observation. GloVe and Tok2Vec obtain similar distances with the former performing slightly better. The distances were 9.3 and 10.5, respectively. Note that for this experiment, only words that occur at least twice were considered, which is different from results presented in Sec. 2.1. The mean distance of the model with no embeddings was 35.4, which was considerably higher. Additionally, there was a much larger standard deviation for this model, which suggests that it was randomly guessing.

2.3 Plots of EQ Parameters

In this section, an error analysis of predictions made by the machine learning models was performed. Individual test words are looked at to investigate whether the neural network is actually learning semantic meanings. HQ words are predominantly looked at because they are common in the audio mixing literature and would be more intuitive to evaluate. In Figs. 3 and 4, the EQ settings of human labels are plotted alongside the predictions of Tok2Vec, Glove-840B, and No Embedding. Because the literature does not comprise an "ideal" metric for the task of predicting EQ parameters, graphs were plotted, and the predictions of the algorithms were actually visualized. Fig. 3 plots the graphs for words selected from test folds 1 and 2. Fig. 4 plots the graphs for words selected from test folds 3 and 4. Note that for each word in the test folds, the neural network has not encountered the word in the training set. The human label chosen for each semantic word in the plots was the EQ setting with the highest consistency score in the dataset.

In Fig. 3, human labels for *muffled* had boosts at 20 Hz and 3.5 kHz. For Tok2Vec and GloVe, slight boosts were seen in the mid-range and high-range, respectively, which may convey that the neural networks did not interpret this word correctly. The predictions made by Tok2Vec and GloVe were also observed to be considerably different from each other. This can be due to two reasons—(1) Tok2Vec and GloVe are different algorithms and therefore learn different semantic meanings from text, and (2) there may be a higher degree of randomness in their predictions because the embeddings are trained only on natural text from the Word Wide Web, which is different from EQ descriptors. Hence, the neural network would require more training examples containing EQ descriptors. The network with No Embedding was basically a flat curve for all the words in the first fold.

For *crisp*, interestingly, the predictions of Tok2Vec and GloVe did follow a similar pattern as the human labels. In the human labels, boosts were seen at 2,100 and 9,000 Hz. For GloVe and Tok2Vec, a gradual boost was seen at 3,000 Hz, which lifts the high-range of the frequency spectrum. Some semantic synonyms of *crisp* present in the training set for this respective fold include *bright*, *harsh*, *hollow*, and *sharp*. This means that the word embedding has delineated a relationship between the semantic word and EQ predictions. Again, as mentioned earlier, a meaningful pattern in the neural network with No Embedding was not observed because the curves were flat.

Muddy had a gradual boost from 200 to 380 Hz in the human labels. Tok2Vec follows a very similar pattern in its prediction by boosting the lows and cutting the highs. GloVe's prediction has slightly boosted lows and highs, which is not convincing for the semantic word *muddy*. Some semantic synonyms in the training set include *boom*, *muddled*, *dark*, *dull*, and *fat*. The next test word, *brittle*,

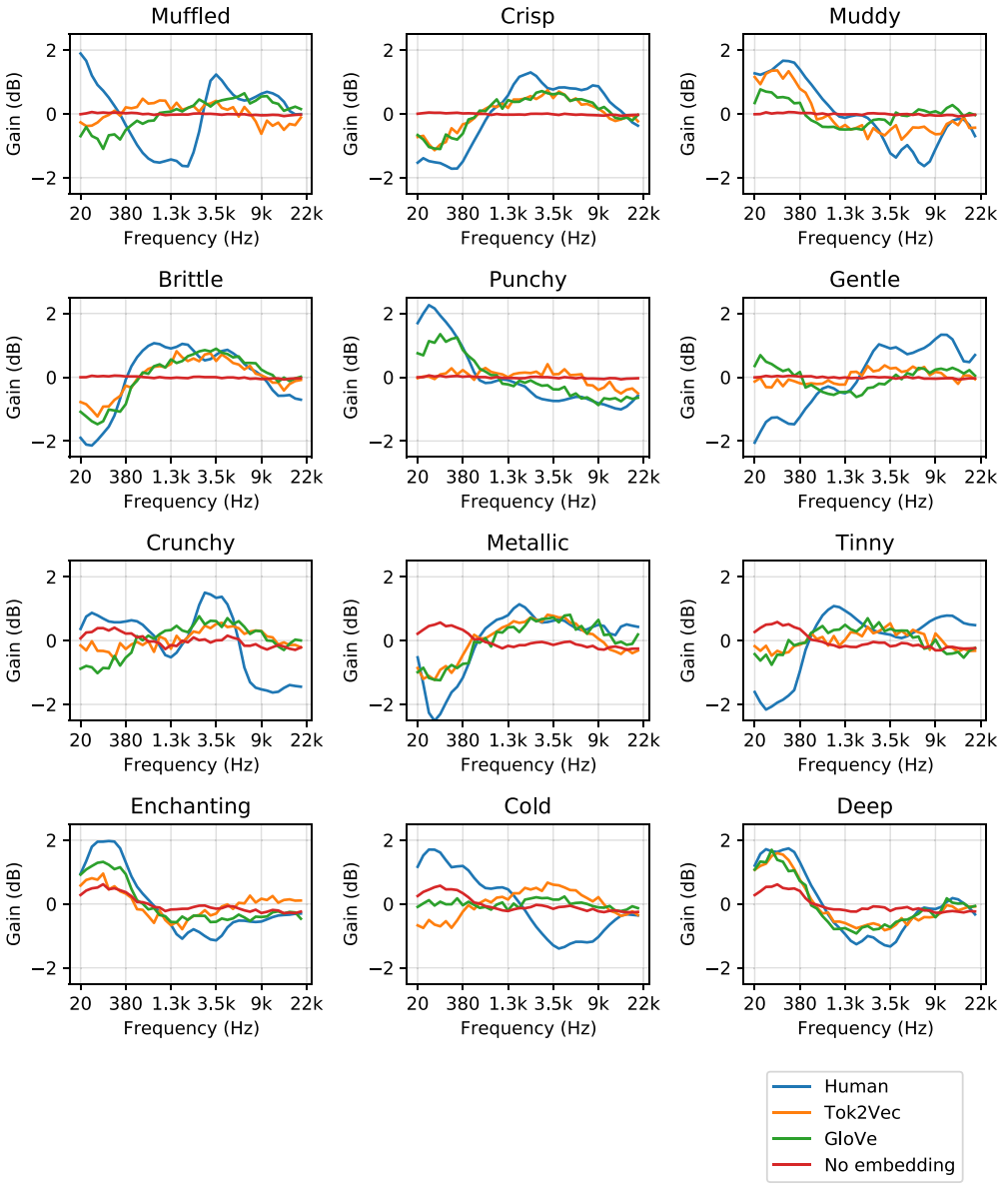

Fig. 3. Plots of equalizer (EQ) parameters for words in test folds 1 and 2. Note that each word in the test set does not occur in the training set. The first two rows occur in fold 1, and the last two rows occur in fold 2. The human label plotted for a semantic word was the EQ settings with the highest consistency score in the dataset.

was well-understood by both Tok2Vec and GloVe. There was considerable overlap with the human labels. The synonyms for *brittle* in the training set would be similar to those listed for *crisp*. *Punchy* was understood by GloVe but not by Tok2Vec. *Gentle* was not understood by either embedding model. (Please refer to Table 1 for more semantic synonyms. If fold 1 is selected as the test set, folds 2, 3, and 4 are included in the training set.)

Crunchy had boosts in the low-frequency and high-frequency ranges in the human labels. A boost for GloVe and Tok2Vec is observed in the high range. The No Embedding model has a boost in the low range. However, if you observe, it has made the same prediction for all the test words in the second fold. GloVe and Tok2Vec and correctly understood the semantic descriptor *metallic* and have significant overlap with human labels. For *tinny*, human labels have boosts at 1,300 and 9,000 Hz, whereas the neural networks with embeddings have a gradual boost around 3,000 Hz. It is not certain whether these predictions would have a *tinny* effect. For test words *enchanting* and *deep*, a noticeable overlap with human labels was observed. However, for *cold*, it seems as though GloVe and Tok2Vec predicted the antonym.

In test fold 3, *sweet* was not understood by the networks at all. For *warm*, Tok2Vec has a noticeable overlap with the human labels because both have a boost of approximately 2 dB in the low-frequency range. *Airy* was partially convincing because GloVe recognized a boost at 9 kHz. Although the networks have boosted the lows for *full*, it seems like a random guess because the prediction significantly overlaps with the one made by No Embedding. The predictions made by the networks for *boxy* were not convincing. *Bright* seemed plausible with Tok2Vec and GloVe boosting the high-frequency range.

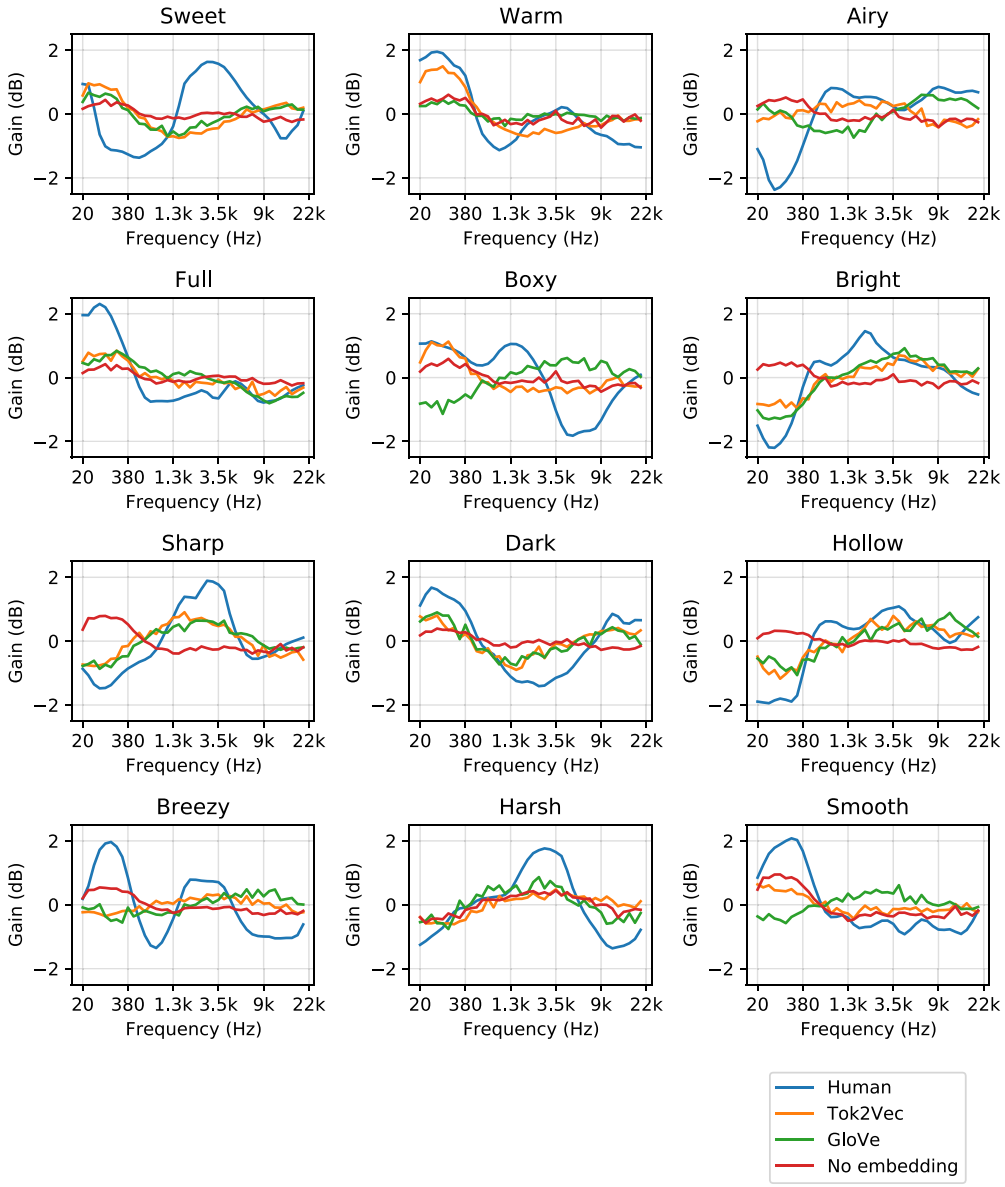

Fig. 4. Plots of equalizer (EQ) parameters for words in test folds 3 and 4. The first two rows occur in fold 3, and the last two rows occur in fold 4.

In test fold 4, reasonable overlap for *sharp*, *dark*, *hollow*, and *harsh* was seen. A reasonable pattern for *breezy* and *smooth* was not observed. Interestingly, the network with No Embedding predicted the EQ settings for *harsh* correctly. This is a chance occurrence because it predicted a standard template of settings for all the other words.

In Fig. 5, the predictions on non-technical words are analyzed. These non-technical words are the same as the HR words explained in Sec. 0. Although these words may have a high consistency score in the SocialFX dataset, they may be highly subjective to the user. However, the predictions of GloVe and Tok2Vec were compared to the human labels. There was considerable overlap for *mellow*, *enchanting*, *rich*, and *romantic*. For *mournful* and *calm*, there were similar patterns between the predictions of the word-embedding models and human labels. However, for *heat* and *brass*, the word-embedding models did not predict a relevant pattern. Although the training set contained semantically similar words like *warm* and *brassy*, the embeddings did not perceive these similarities. This conveys that the algorithms to learn word embeddings can be further optimized for EQ mixing.

3 DISCUSSION

In the previous section, a word-embedding layer was shown to be helpful for automatic mixing. The error of models in Sec. 2.1 was analyzed. All the models with an embedding layer obtained lower errors than the one without an embedding layer. The performance of GloVe-840B and Tok2Vec was further analyzed by using PCM. The mean distances obtained by human labels, Tok2Vec, GloVe, and No Embedding were 2.9, 10.5, 9.3, and 35.4, respectively. This objectively demonstrates that the embedding models

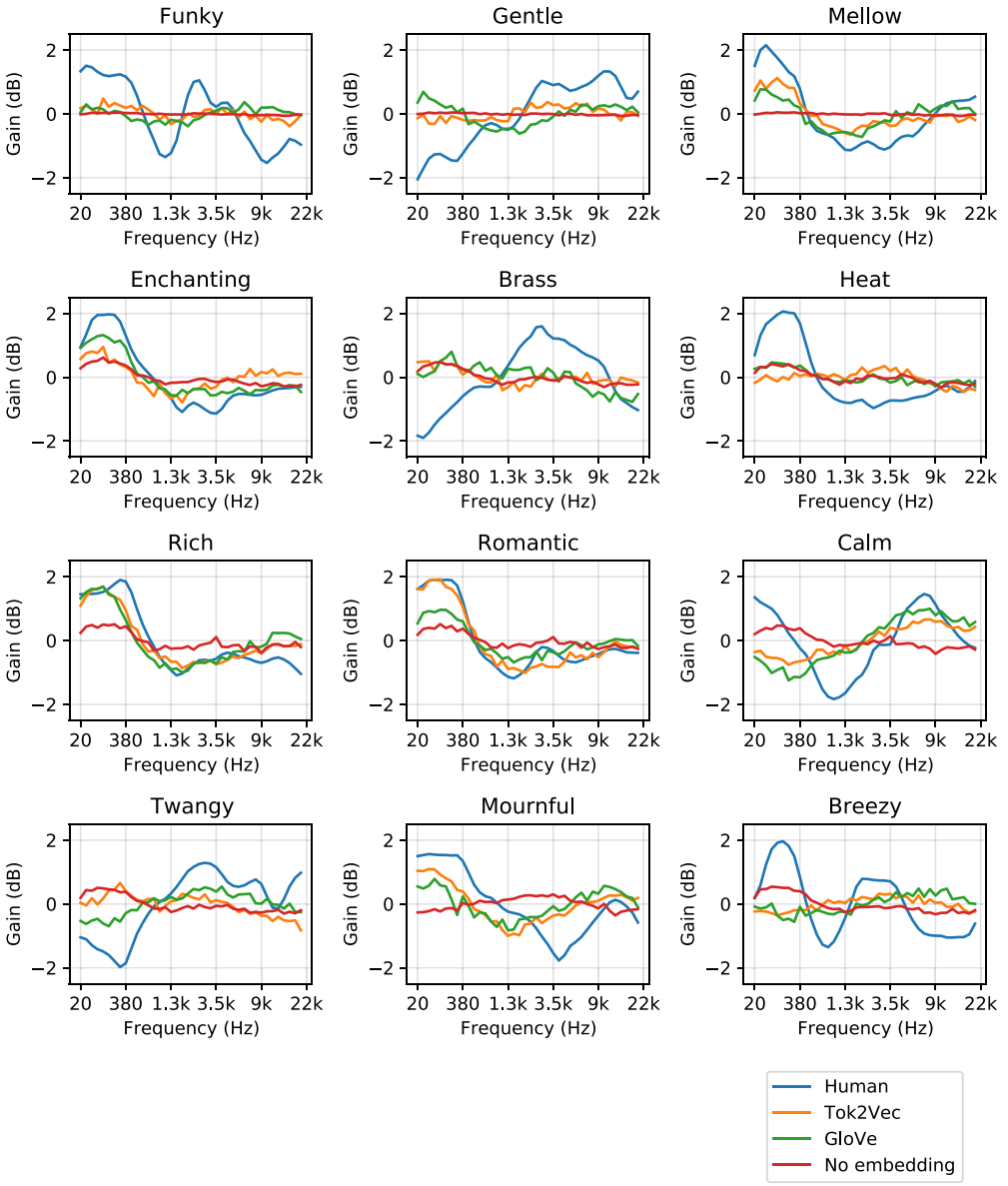

Fig. 5. Plots of equalizer (EQ) parameters for highly rated (HR) words as explained in Sec. 1.2. These are non-technical words that may be highly subjective to a user.

perform better than models without an embedding layer but not as good as human labels.

In Sec. 2.3, an error analysis of predictions made by GloVe and Tok2Vec was conducted. It was observed that the machine learning models were able to understand semantic descriptors that they had not encountered before. This is a promising step toward understanding semantic descriptors from non-technical users. It is important to note the word-embedding layers used in the networks were trained on corpora of written text. This concludes that there exists some common ground for semantic relationships between words in written text and for those adopted in EQ mixing.

Considering the fact that such a small training dataset has been adopted, this performance is reasonable. The SocialFX dataset comprises only 388 unique English words. Additionally, many of the HQ and HR words were used for testing in each fold. Because this study has demonstrated that word embeddings are helpful for automatic EQ mixing, the authors hope to encourage researchers to build larger datasets with semantic descriptors. In the literature, another dataset called SAFE [20] focused on extracting semantic descriptions for equalization from a DAW. The dataset was not included within this study for two reasons. Firstly, because these are extracted directly from the DAW without post-processing, some labels can be noisy. Although the dataset contains many examples with meaningful descriptors, some words are randomly typed letters such as “xy,” which have no semantic meaning. Perhaps this noise may not matter when training the network with large-scale data. The second reason is that both datasets use different EQ plugins. The SocialFX dataset uses a 40-band EQ, whereas the SAFE dataset uses a five-band EQ. It is not certain whether additional noise would be induced in mapping one EQ domain to the other.

In this study, the performance of the machine learning model was analyzed using objective metrics. However, it is important to perform listening tests with human participants to obtain subjective evaluations of the system. Whether users are satisfied with the way the machine learning model understands their semantic descriptors needs to be investigated. After aggregating a larger dataset for this task, this could be a potential future pathway.

4 CONCLUSION

In this paper, the feasibility of adopting word embeddings for automatic EQ mixing was demonstrated. It was shown that the word-embedding layer is capable of providing relationships between semantic descriptors, which assists in predicting EQ parameters. Using this technique, the machine learning model can predict EQ settings for words it has not seen before. This is a step toward bridging the gap between artists explaining their creative goals and mixing engineers understanding them.

In this study, EQ parameters were looked at as a separate entity. This may not be ideal in some scenarios. For example, the EQ settings for "*make the vocals sound brighter*" maybe different from "*make the drums sound brighter*." Moreover, the number of EQ bands predicted was 40. This number is large for a network that performs regression. Future research could explore how the neural network architecture can be optimized and regularized better. Furthermore, it may be interesting to augment the size of training sets by adopting well-known synonyms and antonyms in the mixing engineer's vocabulary.

For some words, Tok2Vec captured relationships, but GloVe did not, and vice versa. For example, GloVe captured the meaning of *punchy* as shown in Fig. 3, and Tok2Vec captured the meaning of *warm* as shown in Fig. 4. This may be simply because there is limited data in the training set. Otherwise, different embedding models may capture different aspects of semantic relationships. Therefore, an ensemble of different embedding models will improve performance in this case. Furthermore, in this study, non-English words were discarded for simplicity. Word-embedding models such as ConceptNet [29] use a knowledge graph to connect words from different languages. This may be an interesting avenue to explore.

5 ACKNOWLEDGMENT

This paper is supported by the Engineering and Physical Sciences Research Council (EPSRC) Grant EP/S026991/1 RadioMe: Real-Time Radio Remixing.

6 DATA AVAILABILITY STATEMENT

This paper uses the SocialEQ dataset [13], which is openly available. The code associated with this study can be found in this GitHub repository (https://github.com/satvik-venkatesh/word-eq).

7 REFERENCES

[1] T. Wilmering, D. Moffat, A. Milo, and M. B. Sandler, "A History of Audio Effects," *Appl. Sci.*, vol. 10, no. 3, paper 791 (2020 Jan.). http://doi.org/10.3390/app10030791.

[2] M. A. Martínez Ramírez, D. Stoller, and D. Moffat, "A Deep Learning Approach to Intelligent Drum Mixing With the Wave-U-Net," *J. Audio Eng. Soc.*, vol. 69, no. 3, pp. 142–151 (2021 Mar.).

[3] B. De Man and J. D. Reiss, "A Knowledge-Engineered Autonomous Mixing System," presented at the *135th Convention of the Audio Engineering Society* (2013 Oct.), paper 8961.

[4] B. De Man, R. Stables, and J. D. Reiss, *Intelligent Music Production* (Routledge, New York, NY, 2020).

[5] D. Moffat, F. Thalmann, and M. B. Sandler, "Towards a Semantic Web Representation and Application of Audio Mixing Rules," in *Proceedings of the 4th Workshop on Intelligent Music Production* (Huddersfield, UK) (2018 Sep.).

[6] E. Perez-Gonzalez and J. Reiss, "Automatic Gain and Fader Control for Live Mixing," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 1–4 (New Paltz, NY) (2009 Oct.). https://doi.org/10.1109/ASPAA.2009.5346498 .

[7] D. Moffat and M. Sandler, "Machine Learning Multitrack Gain Mixing of Drums," presented at the *147th Convention of the Audio Engineering Society* (2019 Oct.), e-Brief 527.

[8] E. T. Chourdakis and J. D. Reiss, "A Machine-Learning Approach to Application of Intelligent Artificial Reverberation," *J. Audio Eng. Soc.*, vol. 65, no. 1/2, pp. 56–65 (2017 Feb.). http://doi.org/10.17743/jaes.2016.0069.

[9] D. Moffat and M. B. Sandler, "Approaches in Intelligent Music Production," *Arts*, vol. 8, no. 4, paper 125 (2019 Sep.). http://doi.org/10.3390/arts8040125.

[10] E. Tarr, *Hack Audio: An Introduction to Computer Programming and Digital Signal Processing in MATLAB* (Routledge, New York, NY, 2019).

[11] S. Stasis, *Audio Equalisation Using Natural Language*, Ph.D. thesis, Birmingham City University, Birmingham, UK (2018 Jul.).

[12] L. d. G. Kulka, "Equalization—The Highest, Most Sustained Expression of the Recordist's Heart," *Rec. Eng./Produc.*, vol. 3, no. 6, pp. 17–24 (1972 Nov.).

[13] M. B. Cartwright and B. Pardo, "Social-EQ: Crowdsourcing an Equalization Descriptor Map," in *Proceedings of the 14th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 395–400 (Curitiba, Brazil) (2013 Nov.).

[14] G. Bromham, D. Moffat, M. Barthet, A. Danielsen, and G. Fazekas, "The Impact of Audio Effects Processing on the Perception of Brightness and Warmth," in *Proceedings of the ACM Audio Mostly Conference: A Journey in Sound*, pp. 183–190 (Nottingham, UK) (2019 Sep.).

[15] T. Zheng, P. Seetharaman, and B. Pardo, "Socialfx: Studying a Crowdsourced Folksonomy of Audio Effects Terms," in *Proceedings of the 24th ACM International*

Conference on Multimedia, pp. 182–186 (Amsterdam, The Netherlands) (2016 Oct.).

[16] P. Seetharaman and B. Pardo, "Crowdsourcing a Reverberation Descriptor Map," in *Proceedings of the 22nd ACM International Conference on Multimedia*, pp. 587–596 (Orlando, FL) (2014 Nov.).

[17] A. Zacharakis, K. Pastiadis, J. D. Reiss, and G. Papadelis, "Analysis of Musical Timbre Semantics Through Metric and Non-Metric Data Reduction Techniques," in *Proceedings of the 12th International Conference on Music Perception and Cognition (ICMPC)*, pp. 1177–1182 (Thessaloniki, Greece) (2012 Jul.).

[18] D. Williams and T. Brookes, "Perceptually-Motivated Audio Morphing: Brightness," presented at the *122nd Convention of the Audio Engineering Society* (2007 May), paper 7035.

[19] E. R. Miranda, "An Artificial Intelligence Approach to Sound Design," *Comput. Music J.*, vol. 19, no. 2, pp. 59–75 (1995 Jun.) http://doi.org/10.2307/3680600.

[20] R. Stables, S. Enderby, B. De Man, G. Fazekas, and J. D. Reiss, "SAFE: A System for Extraction and Retrieval of Semantic Audio Descriptors," Late Breaking/Demo at *the 15th International Society for Music Information Retrieval Conference (ISMIR)* (Taipei, Taiwan) (2014 Oct.).

[21] S. Stasis, R. Stables, and J. Hockman, "Semantically Controlled Adaptive Equalisation in Reduced Dimensionality Parameter Space," *Appl. Sci.*, vol. 6, no. 4, paper 116 (2016 Apr.). http://doi.org/10.3390/app6040116.

[22] E. T. Chourdakis and J. D. Reiss, "Tagging and Retrieval of Room Impulse Responses Using Semantic Word Vectors and Perceptual Measures of Reverberation," presented at the *146th Convention of the Audio Engineering Society* (2019 Mar.), paper 10198.

[23] A. Bakarov, "A Survey of Word Embeddings Evaluation Methods," *arXiv preprint arXiv:1801.09536* (2018).

[24] Y. Goldberg, *Neural Network Methods for Natural Language Processing*, Synthesis Lectures on Human Language Technologies, vol. 10 (Morgan & Claypool, San Rafael, CA, 2017).

[25] G. Forman and M. Scholz, "Apples-to-Apples in Cross-Validation Studies: Pitfalls in Classifier Performance Measurement," *ACM SIGKKD Explor. Newslett.*, vol. 12, no. 1, pp. 49–57 (2010 Jun.). http://doi.org/10.1145/1882471.1882479.

[26] A. Pearce, T. Brookes, and R. Mason, "Hierarchical Ontology of Timbral Semantic Descriptors," Tech. Rep. D5.1 (2016 Aug.).

[27] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global Vectors for Word Representation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543 (Doha, Qatar) (2014 Oct.). http://doi.org/10.3115/v1/D14-1162.

[28] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *arXiv preprint arXiv:1301.3781* (2013).

[29] R. Speer and J. Lowry-Duda, "ConceptNet at SemEval-2017 Task 2: Extending Word Embeddings With Multilingual Relational Knowledge," in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval)*, pp. 85–89 (Vancouver, Canada) (2017 Aug.). http://doi.org/10.18653/v1/S17-2008.

[30] J. Tissier, C. Gravier, and A. Habrard, "Dict2vec: Learning Word Embeddings Using Lexical Dictionaries," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 254–263 (Copenhagen, Denmark) (2017 Sep.). http://doi.org/10.18653/v1/D17-1024.

[31] M. Honnibal, I. Montani, S. Van Landeghem, and A. Boyd, "spaCy: Industrial-Strength Natural Language Processing in Python," *Zenodo* (2020 Nov.). http://doi.org/10.5281/zenodo.1212303.

[32] K. Witowski and N. Stander, "Parameter Identification of Hysteretic Models Using Partial Curve Mapping," in *Proceedings of the 12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, paper 5580 (Indianapolis, IN) (2012 Sep.). http://doi.org/10.2514/6.2012-5580.

[33] C. F. Jekel, G. Venter, M. P. Venter, N. Stander, and R. T. Haftka, "Similarity Measures for Identifying Material Parameters From Hysteresis Loops Using Inverse Analysis," *Int. J. Mater. Form.*, vol. 12, pp. 355–378 (2019 May). http://doi.org/10.1007/s12289-018-1421-8.

THE AUTHORS


Satvik Venkatesh


David Moffat


Eduardo Reck Miranda

Satvik Venkatesh holds a Bachelor of Technology in Information and Communication Technology from SASTRA University, India, and Master of Research in Computer Music from the University of Plymouth, UK. He is currently pursuing a Ph.D. on the topic of audio segmentation and intelligent mixing for live radio broadcast. His research interests include deep learning, brain-computer interfaces, and unconventional computing for music. Satvik is also an accomplished musician and performer.

•

David Moffat is an applied Artificial Intelligence (AI) researcher at Plymouth Marine Laboratory. Previously, he was a Lecturer in Sound and Music Computing at the University of Plymouth. He received his Ph.D. from Queen Mary University in sound synthesis, machine learning, and subjective evaluation. His primary research interests are in the field of intelligent and assistive mixing and audio production through the implementation of semantic tools and AI.

•

Eduardo Reck Miranda is a composer and Artificial Intelligence (AI) scientist working at the crossroads of biology and music. He received a Ph.D. on the topic of musical composition with AI from the University of Edinburgh. Currently, he is a Professor in Computer Music at the University of Plymouth, where he leads the Interdisciplinary Centre for Computer Music Research, which is pioneering the fields of Music Neurotechnology and the development of biological and quantum computing for music.