UNSUPERVISED TEXT-TO-SOUND MAPPING VIA EMBEDDING SPACE ALIGNMENT

Luke Dzwonczyk and Carmine-Emanuele Cella

Center for New Music and Audio Technologies (CNMAT)
University of California, Berkeley
Berkeley, CA, USA
dz.luke@berkeley.edu

ABSTRACT

This work focuses on developing an artistic tool that performs an unsupervised mapping between text and sound, converting an input text string into a series of sounds from a given sound corpus. With the use of a pre-trained sound embedding model and a separate, pre-trained text embedding model, the goal is to find a mapping between the two feature spaces. Our approach is unsupervised which allows any sound corpus to be used with the system. The tool performs the task of text-to-sound retrieval, creating a soundfile in which each word in the text input is mapped to a single sound in the corpus, and the resulting sounds are concatenated to play sequentially. We experiment with three different mapping methods, and perform quantitative and qualitative evaluations on the outputs. Our results demonstrate the potential of unsupervised methods for creative applications in text-to-sound mapping.

1. INTRODUCTION

Natural language processing (NLP) techniques, such as word2Vec and RoBERTa, have demonstrated the ability to capture semantic relationships between words [1, 2]. Similarly, deep learning models for audio analysis have been used to extract meaningful sound representations [3, 4]. However, aligning these two embedding spaces within a unified framework remains an open challenge. This research explores unsupervised learning techniques to address this issue, utilizing their flexibility to bypass the need for labeled audio data, which can be scarce or vulnerable to subjectivity in labeling [5]. By doing so, this work aims to provide a novel computational tool for artistic expression and sound-based interpretation of textual input.

In this system, the user provides a sound corpus along with a text input and the output of the system is a sound file in which each input word has been associated with one sound from the corpus. The collected sounds then play in the order of the words in the input text. The generated sound should not only reflect the connections between words but also align with the characteristics of the chosen sound dataset. Rather than establishing direct semantic correspondences between individual words and sounds, the focus is on preserving relational structures and ensuring that the relationships between words are mirrored in the relationships between their corresponding sounds.

Consider an input text that contains the words “red” and “blue.” Our goal is not that the word red maps to a sound that is somehow “red”, but that the relationship between the “red” sound and the “blue” sound reflects the relationship between the semantic meaning of the two words. As two words that belong to the same category, the resulting sounds should also come from the same category or cluster of sounds. However since the colors red and blue are complementary colors that fall on opposite sides of the color wheel, the two sounds should be distinct in some perceptual or musical feature. A theoretical relation between the sounds could be that they are samples from the same instrument but are different in pitch, exemplifying the similarity in semantic category but difference in characteristic.

The choice of unsupervised methods is crucial to this work for multiple reasons. First, supervised audio and music tasks can be problematic due to the subjectivity of labeled data; what one person considers a "harsh" sound may be "pleasant" to another’s ears [6]. This subjectivity can be especially present across different cultures and musical practices and is compounded when emotionally valenced labels are used [7]. Similarly, genre tagging for music can fall prey to the same issue as many musics fall across multiple genres or evade genre labels altogether [8]. Finally, the use of unlabeled audio data allows a user to provide any type of sound corpus without restraint on the type or variety of sounds.

In summary, our goal is to design a creative tool that:

1. Maps a sound corpus and a text input into distinct embedding spaces using pre-trained models
2. Creates an unsupervised mapping between the two embedding spaces
3. Generates an output sound that is a result of concatenating the mapped sound from each word in the text input

The structure of the paper is as follows: we provide an overview of the state-of-the-art in sound and text embedding models and alignment techniques in Section 2. In Section 3 we present our system with three mapping methods and three evaluation metrics. We explain our experimental setup and provide our results in Section 4, which shows that the simplest mapping method performs the best quantitatively. In Section 5 we comment on the outcomes of our experiments and emphasize the importance of qualitative evaluation for artistic tools. Finally, we conclude and present future avenues to expand the work in Section 6.

Our code is available at: `https://github.com/dzluke/DAFX2025`. You can listen to a selection of generated sounds at the following website: `https://dzluke.github.io/DAFX2025/`.

2. RELATED WORK

2.1. Sound Embeddings

Sound embedding models convert raw audio signals into fixed-dimensional vector representations, enabling downstream analysis

and processing of sound data for a variety of tasks. These models can be broadly categorized into speech-focused models, general audio models, and music-specific models. Speech models such as HuBERT specialize in learning linguistic representations from speech signals, making them effective for tasks like automatic speech recognition [9]. General audio models like ESResNeXt capture broad sound characteristics and are widely used in audio classification and retrieval [10]. Music-oriented models, including MuQ, EnCodec, and MERT, are designed to encode musical structures and enable applications such as music tagging, generation, and retrieval.

Speech embedding models have seen significant advancements, particularly with self-supervised learning techniques. HuBERT (Hidden-Unit BERT) learns representations directly from raw waveforms by predicting masked speech units derived from an offline clustering process [9]. This allows it to capture both phonetic and linguistic features that improve speech recognition and spoken language processing. While highly effective in speech-related tasks, such models are not designed to capture the broader range of non-verbal audio characteristics required for general sound understanding, nor are they suited for musical tasks.

General audio models such as ESResNeXt focus on learning embeddings from a wider variety of sounds, including environmental noise, musical tones, and human speech. ESResNeXt, a convolutional neural network-based model, is trained on large-scale datasets making it well-suited for sound event classification and multimodal retrieval [10].

More appropriate to our task, music embedding models are tailored to capture melodic, harmonic, and rhythmic structures. MuQ, which utilizes Mel Residual Vector Quantization, is designed for tasks like music tagging and instrument classification by learning compact yet expressive music representations [11]. EnCodec, a neural audio codec, compresses and reconstructs high-fidelity audio signals using an encoder-decoder structure with residual vector quantization, producing embeddings that preserve fine-grained musical details [12]. MERT (Music undERstanding model with large-scale self-supervised Training) leverages large-scale self-supervised learning to capture the pitched and tonal characteristics of music, making it particularly effective for music classification and understanding [13]. Unlike speech and general audio models, music embeddings must capture complex temporal structures and timbral variations, making them uniquely suited for applications in composition, generation, and audio synthesis.

2.2. Text Embeddings

Text embedding models such as word2vec, fastText, GloVe, BERT, and T5 have significantly improved how machines represent and process language by encoding words and sentences into high dimensional feature spaces [1, 14, 15, 16, 17] . These models can perform a wide range of tasks including semantic similarity analysis, text classification, machine translation, and information retrieval. Static embeddings like Word2Vec and GloVe capture general word relationships based on co-occurrence patterns, while contextual models like BERT and T5 generate dynamic representations that account for the surrounding context.

Static embeddings, such as Word2Vec, GloVe, and FastText, assign a fixed vector representation to each word regardless of its context in a sentence. These models capture general semantic relationships between words based on large-scale co-occurrence patterns in text corpora. However, they cannot differentiate between words with multiple meanings, as the same embedding is used in all contexts.

Contextual embeddings, such as RoBERTa and T5, generate word representations that depend on the surrounding context in which the word appears; therefore the same word may have different embeddings depending on where it appears in a sentence. These models leverage deep neural networks to dynamically adjust word embeddings based on sentence structure and meaning. This allows them to capture nuanced relationships and word dependencies, making them more suitable for complex language understanding tasks. RoBERTa is a transformer trained with self-supervision on unlabeled text data [2]. It uses masking during training, meaning it learns to predict masked words in a given input sequence, and in that way learns word representations that can be used for a variety of downstream tasks.

2.3. Multi-modal Embeddings

Our task is essentially text-to-sound retrieval, which is a specific instance of the task of cross-modal retrieval. Various multi-modal models that establish a shared feature space for both audio and text exist, allowing for either modality to be input to the space.

CLAP (Contrastive Language-Audio Pretraining) employs a contrastive learning framework to align audio and text representations [18]. It uses a SWINTransformer to extract audio features from log-Mel spectrograms and a RoBERTa model for text features, projecting both into a common latent space. CLAP can be used for downsteam musical tasks, such as applying effects to sound using natural language [19].

AudioCLIP extends CLIP, a text-image model, by incorporating audio, aligning it with text and images in a shared embedding space [20]. Utilizing ESResNeXt for audio encoding, AudioCLIP enables cross-modal retrieval and classification tasks, effectively bridging audio, text, and visual content.

MuLan employs contrastive learning to align music and text embeddings, enabling applications like music retrieval based on textual descriptions [21]. MuQ-MuLan builds upon this by integrating MuQ's music representations, enhancing tasks such as zero-shot music tagging and music-text retrieval [11].

While these models effectively align audio and text based on semantic content, their focus on objective accuracy, such as associating the word "bird" with the sound of birdsong, is not a requirement of of our system. Associating words with sounds based on abstract relationships extends beyond objective semantic alignment, presenting challenges for existing models. We seek a tool that is more flexible and open in the hopes of creating more creative and artistically interesting mappings.

2.4. Alignment Techniques

Our task is to map two disparate feature spaces; one possible solution is to attempt to align the two spaces. This can be achieved through a variety of methods that can be reduced to the idea of a mapping matrix or function that transforms the points of one space to align with the points of another space [22].

In [23], the authors draw an analogy between this problem and the alignment of 3-D point clouds: if we can translate one point cloud to match the distribution of another, we have created an effective mapping between the point clouds. This is essentially Procrustes problem. In the domain of point cloud matching, two common methods for achieving this alignment are the Procrustes

method and the Iterative Closest Point method. The authors of [23] go on to introduce Mini-Batch Cycle Iterative Closest Point, which we call ICP for simplicity, a method for aligning text and speech sounds in an unsupervised way. The algorithm for ICP is as follows:

Given a sound space S and text space T, the first step of ICP is to perform Principal Component Analysis (PCA) to reduce the spaces to the same dimension. Consider two transformational matrices W_S which maps from S to T and W_T which maps from T to S; each matrix is initialized to the identity, under the assumption that the spaces share a basic similarity.

Now perform an iterative process on mini-batches of S and T:

1. For each $s \in S$, find t^*, the nearest $W_T t$ to s, which is the nearest text encoding to s
2. For each $t \in T$, find s^*, the nearest $W_S s$ to t, which is the nearest sound encoding to t
3. Optimize W_S and W_T using mini-batch Stochastic Gradient Descent by minimizing:

$$\sum_j \|s_j^* - W_t t_j\| + \sum_i \|t_i^* - W_s s_i\| + \lambda \sum_i \|s_i - W_T W_S s_i\| + \lambda \sum_j \|t_j - W_S W_T t_j\| \quad (1)$$

The second two terms represent cycle constraints, ensuring that transforming a text to a sound and back to a text would yield the original text. After a number of iterations, we have generated two matrices W_S and W_T which can map between the two spaces [24].

Another alignment technique takes advantage of Generative Adversarial Networks (GANs) to train a mapping between text and sound spaces [25, 26]. The generator in the adversarial game is the mapping matrix W_T, which creates embeddings in the sound space given an embedding in the text space. The discriminator attempts to distinguish between a real sound embedding $s \in S$ and a "fake" sound embedding $W_T t$ that is an output of the generator. After adversarial training, W_T can be used to map text to sound.

3. METHODOLOGY

In order to map text embeddings to sound embeddings, we develop the following system that takes as input a sound corpus and a text string. First, each sound in the corpus is pre-processed, during which it may be sliced into multiple chunks, leading to a total number of sounds n that may be greater than the number of sound files in the corpus. Then, each of these processed sounds are embedded into the sound space, creating a matrix $S \in R^{n,p}$ where p is the output dimensionality of the sound encoder. The input text of m words is similarly embedded into the text feature space of dimensionality q, creating a matrix of text embedding $T \in R^{m,q}$.

Next, both S and T are normalized and dimensionality reduction is performed to reduce them to the same number of features. Our new S and T are of shape (n, d) and (m, d), respectively.

A mapping strategy is employed in order to find a sound embedding that best fits a given text embedding. We use three distinct mapping strategies:

1. **Identity Mapping:** The simplest mapping in which the mapping matrix is the identity matrix, $W_T = I$. For a given text embedding, the sound that is closest to the text embedding is selected as the output.
2. **Cluster Mapping:** First, the text embeddings and clustered and the sound embeddings are clustered. Then, for a given text embedding, the sound cluster whose centroid is closest to the cluster the text embedding is in is selected. Finally, the sound embedding in that cluster that is closest to the text embedding is selected as the output.
3. **Mini-Batch Cycle Iterative Closest Point (ICP):** This method iteratively aligns the text and sound spaces by optimizing transformation matrices to minimize distance and enforce cycle consistency. A mapping matrix W_T is learned through optimization. For a given text embedding t, the nearest sound embedding to $W_T t$ is selected as the output.

The system is evaluated using three distance metrics which are derived from the following logic: given two text embeddings t_i and t_j, and the two corresponding sounds they map to, s_i and s_j, the distance between t_i and t_j should be similar to the distance between s_i and s_j. In this way, words that are similar should map to sounds that are similar, words that are distantly related should map to sounds that are distantly related. From this idea we use the following three distance metrics:

1. **Pairwise Distance:** The distance between text is calculated in the text embedding space T and the distance between sounds is calculated in the sound embedding space S. The total distance is calculated as the sum of the difference between the distances:

$$D = \frac{2}{m(m-1)} \sum_{i,j}^{m} |d(t_i, t_j) - d(s_i, s_j)| \quad (2)$$

   where m is the number of words in the text input, $t \in T$ is a text embedding, $s \in S$ is a sound embedding, and d is any distance function between two embeddings.
2. **Wasserstein Distance:** Instead of subtracting the distances between text and sound pairs, we calculate the Wasserstein distance between the distribution of text pair distances and sound pair distances.
3. **CLAP Distance:** Same as Equation 2 except the distances between text pairs and sound pairs are calculated in the shared embedding space of the CLAP model [18].

When the clustering mapping is performed, we use clustering metrics, including the silhouette score, Calinski-Harabasz score, and Davies-Bouldin score, to evaluate the quality of a clustering result by analyzing the compactness and separation of clusters. The silhouette score measures how similar a data point is to others within its cluster compared to those in neighboring clusters, with values ranging from -1 (poor clustering) to 1 (well-defined clusters) [27]. The Calinski-Harabasz score evaluates the ratio of between-cluster variance to within-cluster variance, with higher values indicating more distinct and well-separated clusters [28]. The Davies-Bouldin score, in contrast, assesses the average similarity between each cluster and its most similar cluster, where lower values suggest better clustering as it indicates minimal overlap between groups [29]. These metrics are particularly useful for datasets where the inherent cluster structure is unknown, providing a quantitative means to compare different clustering approaches and parameter choices.

4. EXPERIMENTS AND RESULTS

We consider one experiment to be a unique setting of the following parameters: sound corpus, text input, sound embedding method, text embedding method, sound pre-processing method, normalization method, number of PCA components, mapping method, distance metric for nearest neighbor search, and number of clusters, if the clustering method is used. In total, we ran 7,346 experiments across 6 sound corpora and 3 text inputs.

We experiment with a single sound embedder, MuQ, using the pre-trained *MuQ-large-msd-iter* model, which embeds a given sound as a matrix of shape (t, 1024) where t is relative to the length of the sound file. In order to adjust this embedding to return a matrix of consistent shape, we average across time to return a vector of length 1024. Therefore, our sound embeddings exist in 1,024-dimensional space.

Three text embedders are tested: word2vec, fastText, and RoBERTa. For word2vec, we use the *word2vec-google-news-300* pre-trained model, which is trained on 3 million words from the Google News dataset. For fastText we use the *cc.en.300* pre-trained model, which is trained on Common Crawl and Wikipedia [30]. Both models embed words in a 300-dimensional space.

As previously mentioned, word2vec and fastText are static embedders in which the position of a word in a sentence is not considered. These models return the same embedding for an input word every time. fastText has the advantage of being able to process words from outside the dictionary it was trained on, whereas word2vec cannot. If word2vec encounters a word it has not seen during training, it will skip the word and there will not be an associated sound for that word in the final sound file.

RoBERTa is a contextual model, meaning the same word can result in different embeddings based on its position and context in the input sequence. We use the *roberta-base* pre-trained model hosted on HuggingFace which embeds words in a 768-dimensional space. One feature of RoBERTa is the ability to understand rare or complex words by splitting them into sub-words that it can properly parse. In order to maintain a one-to-one mapping between words and sounds, if a word is split into sub-words we average the embeddings for each sub-word and use this as the embedding for the word.

We implement three sound pre-processing methods. For each method, there is the option to remove silence from the input sound before applying further processing. The selected method is run on each sound in the provided corpus, and the outputs of the method are input to the sound embedding model. The three methods allow the input sounds to be chunked in different ways:

1. **full**: The entire input sound is used, no chunking is performed
2. **onsets**: An onset detection is performed and the sounds are chunked at the beginning of a new onset
3. **grain**: Given a grain size g in milliseconds, the input sound is chunked into equal length grains of length g

For onset detection, we use librosa's `onset.onset_detect` method [31].

We use sklearn's StandardScaler as our normalization method, which subtracts the mean and divides by the standard deviation [32]. For dimensionality reduction, sklearn's PCA with default settings is used with number of components set to 2, 5, 10, and 20. We test all three mapping methods defined in Section 3, which we call the identity, cluster, and ICP methods. We use Euclidean distance and cosine distance as our two distance metrics, which are used to find the nearest neighbor and as the distance functions used in our mapping evaluations. For clustering, we use sklearn's implementation of KMeans and test with the number of clusters k set to 2, 5, 10, 20, and 30.

For the implementation of ICP, we perform a grid search to find the optimal hyper-parameters and settle on the following values: a learning rate of 0.001, batch size of 16, cycle weight $\lambda = 0.7$, with 75 iterations per batch.

4.1. Evaluation

As detailed in Section 3, we use three distance metrics to evaluate each experiment: pairwise distance, Wasserstein distance, and CLAP distance. For calculating the Wasserstein distance, we use scipy's `scipy.stats.wasserstein_distance` [33]. For calculating the CLAP distance, we use the *laion/clap-htsat-fused* pre-trained model from HuggingFace which embed sound and text in a 512-dimensional space.

For experiments that use clustering, we evaluate the clustering with three metrics: silhouette score, Calinski-Harabasz score, and Davies-Bouldin score, all of which are implemented with sklearn's `metrics` library. After the sound and text spaces are clustered separately, we create a combined space that contains the clusters of both spaces, and evaluate the clustering of that combined space. We calculate the Pearson's correlation coefficient between each mapping evaluation and each clustering metric in order to discover the relationship between mapping quality and clustering quality [34].

4.2. Quantitative Results

We present a comparison of each mapping method (identity, cluster, and ICP) and the average values for each mapping evaluation method (pairwise, Wasserstein, and CLAP) in Table 1. In Table 2, we show both the mapping evaluations and clustering metrics to compare how the number of clusters k affects the result. This also demonstrates the relationship between clustering quality and mapping distance.

In order to further understand the relationship between mapping quality and clustering quality, we calculate the Pearson correlation coefficient between each mapping evaluation and each clustering metric as shown in Table 3. This measures the strength and direction of the linear relationship between two variables, where 1 indicates a perfect positive correlation, -1 indicates a perfect negative correlation, and 0 suggests no linear relationship.

Mapping	Pairwise	Wasserstein	CLAP
Identity	**5.40**	**3.80**	**0.411**
Cluster	5.53	3.84	0.426
ICP	5.86	4.08	0.417

Table 1: Comparison of average mapping distance for different mapping methods, using a Euclidean distance metric. Lower values are better; the best values for each metric are bolded.

4.3. Qualitative Results

Qualitative evaluation involved listening tests by the authors to assess the artistic interest of the generated sounds, which is a subjective measure heavily influenced by our listening preferences and

	Mapping Evaluations			Clustering Metrics		
Number of Clusters	**Pairwise** (lower is better)	**Wasserstein** (lower is better)	**CLAP** (lower is better)	**Silhouette** (higher is better)	**Calinski-Harabasz** (higher is better)	**Davies-Bouldin** (lower is better)
2	5.64	4.08	0.428	0.043	161	3.53
5	5.46	3.90	0.429	**0.157**	**373**	1.50
10	5.65	3.80	0.440	0.131	330	1.19
20	5.59	3.78	0.419	n/a	348	0.989
30	**5.28**	**3.54**	**0.407**	n/a	363	**0.911**

Table 2: Comparison of average mapping distance and cluster metrics for different number of clusters k. Bolded values are the best for that metric. The silhouette scores for $k = 20$ and $k = 30$ are purposefully omitted; the score could not be calculated as there were not enough samples per cluster.

	Silhouette (negative is better)	**Calinski-Harabasz** (negative is better)	**Davies-Bouldin** (positive is better)
Pairwise	0.14	-0.37	0.08
Wasserstein	0.17	-0.30	0.05
CLAP	0.22	-0.08	-0.06

Table 3: Pearson correlation coefficients between the mapping evaluation metrics and clustering metrics for experiments that use the cluster mapping method.

histories. While this can be problematic if presented as an objective evaluation, we believe it is still valuable since this is an artistic tool whose value is determined by the preferences of the user.

The choice of sound corpus and text input has the largest effect on the resulting sound. We find that using poetry as input can lead to interesting results, since the use of meter and repetition of words can create a similar sense of meter/rhythm through repetition in the resulting sound. The type of sound pre-processing is also important as it determines the length of the sounds that are concatenated. If the pre-processing results in sound chunks of equal length then the sense of rhythm is further strengthened since new onsets occur at regular periods.

The system is prone to repetition of samples if a static text encoder is used with a text input that features repetition of words. When RoBERTa, a contextual model, is used then repetitions of the same word do not necessarily have the same encoding, leading to less exact repetition in the sound output. For example, in a text input that features the word "and" three times and a selection of singing voices as the sound corpora, fastText chose the exact same sound sample to represent this word each time. However, when RoBERTa was used for the same input, it chose three samples of the same vowel being sung but from different singers, creating a cohesiveness without exact repetition.

We found the sound outputs from the identity and cluster methods to be more similar to each other than the ICP mappings. This is likely due to the fact that ICP attempts to align the two spaces with an additional transformation matrix, which would lead to a different choice of sound than if the transformation had not been applied. This can be useful when the user finds a selection of parameters they like (sounds, text, encoders, pre-processing method) but wants to create a different version of the output, ICP can provide a new mapping that uses different sounds but keeps the rest of the parameters the same. Since ICP is a stochastic method, each run of the method can result in a different mapping, leading to a wide variety in its outputs even for the same input parameters.

5. DISCUSSION

Quantitatively, the identity mapping performs best across all three evaluation metrics, followed by the clustering method, and finally by ICP. Both pairwise distance and Wasserstein distance rank the mapping methods in that order, and CLAP distance ranks identity as the best, followed by ICP and clustering last. The similar ranking across evaluation methods suggest that the methodology of comparing pairwise distances is sound.

The number of clusters k used in clustering has an effect on the evaluation, with higher values of k leading to better mappings. In fact, the clustering method outperforms the identity method if only experiments that use $k = 30$ are considered, which suggests that fine-tuning the number of the clusters for the specific input data could improve the effectiveness of the clustering method. Similarly, we held the hyper-parameters of ICP constant across all experiments but it is possible the ICP mapping could improve if the parameters were optimized for each input.

For the clustering method, we investigate whether the quality of the clustering effects the quality of the mapping. Although the relationship is not clear, the data suggests that a better clustering may lead to a better mapping. This is illustrated by the fact that as k increases, the quality of the mapping increases as represetned by the Davies-Bouldin score, which shows its best value for $k = 30$. This is strengthened by the fact that the second best Calinski-Harabasz score also corresponds to $k = 30$. However, this finding is not supported by the silhouette score, which has the best score at $k = 5$. It was not possible to calculate the silhouette score for $k = 20, 30$ as there cannot be a cluster with less than two samples in order to calculate the silhouette score, and due to the small number of samples in some of our sound corpora and text inputs, with large k values it was not guaranteed that every cluster would have more than two samples.

The Pearson correlation coefficients show mixed results. The Calinski-Harabasz score has the most evidence to suggest that better clustering, as indicated by a higher value of the score, leads to a better mapping. However, the Davies-Bouldin index shows a

near zero correlation across all three mapping evaluations, and the silhouette score shows the a positive correlation, which means a better clustering leads to a worse mapping.

It is surprising that the simplest mapping method performs the best, and that as the complexity of the method increases, the quality of mapping decreases. This could suggest that simple approaches work best for the type of data we input, or that the metrics used favor simple mappings over complex mappings. There could be other reasons for the discrepancy in evaluations, such as the parameters used in the clustering and ICP methods. For the ICP method, the initialization used for mapping matrices is known to be important for the effectiveness of the system [23]. We use the same initialization as the original paper, which is the identity matrix. This choice is a result of the assumption in the original paper that the two spaces have a similarity in distribution as they are both representing language data. In our problem this is not the case, we have both language and non-language data, and it is possible that this assumption is invalid.

In our qualitative evaluation, we did not find a clear ranking in terms of which mapping method results in the best sound outputs. The identity and cluster methods often lead to nearly identical mappings, and ICP would result in a different set of sound outputs for the same text input. Therefore, the choice of mapping methods is not based on objective accuracy but in artistic interest as determined by the user.

6. CONCLUSION AND FUTURE WORK

This research demonstrates the potential of unsupervised methods for text-to-sound mapping, providing a framework that aligns text and sound embeddings to create artistically interesting mappings. The system's flexibility allows it to adapt to various datasets and parameter configurations, making it a valuable tool for creative applications.

We experimented with a variety of sound and text inputs, text encoders, pre-processing methods, and mapping methods. By performing a quantitative evaluation on our outputs we were able to determine which methods and settings led to the "best" result. However, as a creative tool, the objective evaluation is secondary to the opportunities for interesting and varied outputs. Ultimately, these different parameters serve to enable artistic control of the output in different ways, based on the preferences of the user.

There are many paths for continuing and expanding this work. Other mappings and alignment techniques could be attempted. More work can be done to refine and improve the ICP mapping by improving the initialization of the mapping matrices through Optimal Transport or through further fine-tuning after optimization, such as running Procrustes method [23]. For example, the adversarial approach described in Section 2.4 could provide its own mapping or could be the initialization matrix for ICP.

A further analysis of evaluation methods and distance metrics could be useful. We would like to explore the tradeoffs of using Euclidean vs. cosine distances in finding nearest neighbors and calculating our distance metrics. Furthermore, additional analysis in the relationship between clustering quality and mapping quality could shed more light on the effectiveness of the clustering strategy.

An extension of the system could allow for a text corpus to be input that generates the mapping in the same way, but the text that is converted to sound is input separately by the user. This input text could have words not seen in the text corpus, and the system would have to map a new text it has not seen before. This is analogous to training a network and then providing unseen data at test time.

Currently, the system performs the task of text-to-sound retrieval, but a different approach could easily lead to text-to-sound generation. If the sound encoder used creates a continuous feature space in which any point in the space can be sampled to create a coherent sound, then instead of finding the nearest neighbor in the sound space to a mapped text embedding, the system could simply generate the sound that exists at the mapped embedding. This would be possible if the sound encoder is a variational auto-encoder. Unlike current text-to-sound methods which attempt an objective semantic match between input text and output sound (generating the sound of a bird if the input text is "bird"), this system would provide a different lens on the task of text-to-sound that is focused on artistic creation.

As an artistic tool, a more robust qualitative evaluation that includes input from multiple types of users, including sound artists and composers, writers and poets, and non-artists would yield a more well-rounded evaluation. The possibility of a real-time tool in which sounds can be immediately retrieved while the user types could prove interesting, even more so if the sound corpus can be added to and changed in real-time, for example tracking microphone input from performers or instruments. This tool is essentially a monophonic instrument, as it never creates a sound in which two samples play at the same time. A polyphonic version could take each line of text input as a separate voice to be played concurrently, sonifying each line of poetry in a stanza at the same time, for example.

7. REFERENCES

[1] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeffrey Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2013, vol. 26, pp. 3111–3119.

[2] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[3] Hendrik Purwins, Bo Li, Tuomas Virtanen, Jan Schlüter, Shuo-Yiin Chang, and Tara Sainath, "Deep learning for audio signal processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 2, pp. 206–219, 2019.

[4] Aaqib Saeed, David Grangier, and Neil Zeghidour, "Contrastive learning of general-purpose audio representations," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[5] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli, "wav2vec 2.0: a framework for self-supervised learning of speech representations," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2020, NIPS '20, Curran Associates Inc.

[6] Minhee Lee, Seungheon Doh, and Dasaem Jeong, "Annotator subjectivity in the musiccaps dataset," in *HCMIR@ISMIR*, 2023.

[7] JH Lee and X Hu, "Cross-cultural similarities and differences in music mood perception," in *iConference 2014 Proceedings*, 2014.

[8] Romain Brisson and Renzo Bianchi, "Perception of the usability of music-genre labels for the assessment of musical tastes," *Psychology of Music*, vol. 50, no. 4, pp. 1362–1368, 2022.

[9] Wei-Ning Hsu, Alexei Baevski, Abdelrahman Mohamed, and Michael Auli, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.

[10] Andrey Guzhov, Federico Raue, Jörn Hees, and Andreas Dengel, "Esresne(x)t-fbsp: Learning robust time-frequency transformation of audio," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*. 2021, pp. 1–8, IEEE.

[11] Haina Zhu, Yizhi Zhou, Hangting Chen, Jianwei Yu, Ziyang Ma, Rongzhi Gu, Yi Luo, Wei Tan, and Xie Chen, "Muq: Self-supervised music representation learning with mel residual vector quantization," *arXiv preprint arXiv:2501.01108*, 2025.

[12] Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi, "High fidelity neural audio compression," *Trans. Mach. Learn. Res.*, vol. 2023, 2023.

[13] Yizhi Li, Ruibin Yuan, Ge Zhang, Yinghao Ma, Xingran Chen, Hanzhi Yin, Chenghao Xiao, Chenghua Lin, Anton Ragni, Emmanouil Benetos, Norbert Gyenge, Roger Dannenberg, Ruibo Liu, Wenhu Chen, Gus Xia, Yemin Shi, Wenhao Huang, Zili Wang, Yike Guo, and Jie Fu, "MERT: Acoustic music understanding model with large-scale self-supervised training," in *The Twelfth International Conference on Learning Representations*, 2024.

[14] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov, "Enriching word vectors with subword information," in *Transactions of the Association for Computational Linguistics (TACL)*, 2017, vol. 5, pp. 135–146.

[15] Jeffrey Pennington, Richard Socher, and Christopher D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2014, pp. 1532–1543, Association for Computational Linguistics.

[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2019, pp. 4171–4186, Association for Computational Linguistics.

[17] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020.

[18] Benjamin Elizalde, Soham Deshmukh, Mahmoud Al Ismail, and Huaming Wang, "Clap learning audio concepts from natural language supervision," in *2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[19] Annie Chu, Patrick O'Reilly, Julia Barnett, and Bryan Pardo, "Text2fx: Harnessing clap embeddings for text-guided audio effects," 2025.

[20] Andrey Guzhov, Federico Raue, Jörn Hees, and Andreas Dengel, "Audioclip: Extending clip to image, text and audio," in *2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 976–980.

[21] Qingqing Huang, Aren Jansen, Joonseok Lee, Ravi Ganti, Judith Yue Li, and Daniel P. W. Ellis, "Mulan: A joint embedding of music audio and natural language," in *Proceedings of the 23rd International Society for Music Information Retrieval Conference (ISMIR)*, Bengaluru, India, 2022, pp. 559–566.

[22] Saksham Singh Kushwaha and Magdalena Fuentes, "A multimodal prototypical approach for unsupervised sound classification," 2023.

[23] Yedid Hoshen and Lior Wolf, "Non-adversarial unsupervised word translation," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, Eds., Brussels, Belgium, Oct.-Nov. 2018, pp. 469–478, Association for Computational Linguistics.

[24] Yi-Chen Chen, Chia-Hao Shen, Sung-Feng Huang, and Hung yi Lee, "Towards unsupervised automatic speech recognition trained by unaligned speech and text only," 2018.

[25] Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou, "Word translation without parallel data," in *International Conference on Learning Representations*, 2018.

[26] Yu-An Chung, Wei-Hung Weng, Schrasing Tong, and James Glass, "Unsupervised cross-modal alignment of speech and text embedding spaces," in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. 2018, vol. 31, Curran Associates, Inc.

[27] Peter J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Computational and Applied Mathematics*, vol. 20, no. 1, pp. 53–65, 1987.

[28] Tadeusz Calinski and J. Harabasz, "A dendrite method for cluster analysis," *Communications in Statistics*, vol. 3, no. 1, pp. 1–27, 1974.

[29] David L. Davies and Donald W. Bouldin, "A cluster separation measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 1, no. 2, pp. 224–227, 1979.

[30] Common Crawl Foundation, "Common Crawl Corpus," `https://commoncrawl.org`, 2007, Accessed: March 29, 2025.

[31] Brian McFee, Colin Raffel, Dawen Liang, Daniel P. W. Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th Python in Science Conference*, 2015, pp. 18–25.

[32] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, and D. Cournapeau, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[33] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Charles R. Harris, K. Jarrod Millman, Stefan J. van der Walt, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Robert Kern, et al., "Scipy 1.0: Fundamental algorithms for scientific computing in python," *Nature Methods*, vol. 17, no. 3, pp. 261–272, 2020.

[34] Karl Pearson, "Notes on regression and inheritance in the case of two parents," *Proceedings of the Royal Society of London*, vol. 58, pp. 240–242, 1895.

[35] Yusong Wu, Ke Chen, Tianyu Zhang, Yuchen Hui, Taylor Berg-Kirkpatrick, and Shlomo Dubnov, "Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation," 2022.