Speech and Lyric-based Doc2Vec Music Recommendation System

Niharika Gali
Indian Institute of Information Technology Allahabad
Prayagraj, India

Dr. Vineet Tiwari
Department of Management Studies
Indian Institute of Information Technology Allahabad
Prayagraj, India

***Abstract*—Traditionally, music recommender systems usually come with a text-based user interface where you type a song to get similar recommendations. However, it will serve to be more user-friendly and hands-free should there be providence for a speech-based input instead. While there are provisions for going to an app and looking for similar songs or typing out a song and finding similar content, there is yet no option to simply say a song into your phone and get recommendations for similar listens. This paper aims to provide a framework for a speech and lyric-based music recommendation system where the user can speak out a song and in return, get a list of similar songs based on the lyrics of the inputted song. Using a lyric-based system for finding similar songs is the way to go when there is minimal user input and a content-based recommendation system through Doc2Vec embedding provides feasibility for just that.**

Keywords—Doc2vec; content-based recommendation system; speech recognition; natural language processing

I. INTRODUCTION

With the advent of music applications such as Spotify and Apple Music, the world of music is spinning faster than ever before. People don't have the time or patience to search for new songs and find it very convenient for a system to recommend songs to them.

While there are options to go to an application and search for similar songs based to your likings, sometimes it is much more convenient to just pick your phone, say out a song name and get a list of similar songs with listening options. This is an area that hasn't yet been explored fully and is both very user friendly and a feature that is surely going to be a great addition to any phone or laptop.

When it comes to song recommendations, there are two ways to go about. Either a content-based filtering system is used or a collaborative filtering system is used. A content based filtering system will be faster computation wise and is easy to use when there is not much history or past information to make new recommendations [1]. Collaborative filtering systems use a rating matrix and recommend things to users based on what other users with similar interests are liking [2].

For the case of this lyric-based recommendation system where the user is giving out a fresh song to make recommendations from and has not shared any past listening experience, a content-based filtering system is the most appropriate choice.

This paper aims to recommend songs that are similar lyrics wise and that the user might enjoy based on their interests. The whole system works the following way. First, we say a song's name and artist's name out as raw audio. The speech model picks up the audio and gives out the song and artist's name to the lyric-finding model. This model uses a music dataset through an API and sends in the song's name to extract all its lyrics. Then, the vector representation of the song lyrics is calculated and is compared against vector representations of all other song lyrics using a Doc2Vec embedding model. Finally, the similarities between the different songs are calculated using cosine similarity, and similar songs based on lyrics are given as output.

II. LITERATURE REVIEW

Music recommendation systems have been here for a long time, though limited by the technology of their time. In [3], a music recommendation system based on both content-based and collaborative filtering algorithms is used so that the individual benefits of using each filtering system are individually highlighted and the right choice to use them based on the available data can be made.

In [4], a deep speech-based speech recognition system is described that moves away from the traditional norms of speech recognition to increase efficiency. It scraps away from the need for a phoneme dictionary and uses an optimized Recurrent neural network (RNN) to recognize speech. This makes the output much more accurate and also much faster.

Document classification is harder than simple word embedding because of the structure of data and sparse labels. However, methods like Doc2Vec originating from Word2Vec have been shown to be accurate in finding embeddings for paragraphs and documents in a not so complicated and understandable way [5].

While dot product is traditionally used in most neural networks, it is unbounded and thus at the risk of large variance, making the model more sensitive and turbulent. Cosine similarity is used in order to draw a boundary on the dot product and reduce the variance [6]. Additionally, cosine similarity is vastly used to find similarities in word and document analysis.

III. METHODOLOGY

A. Speech to Text

Speech Recognition is one of the most classical machine learning problems. Traditionally, it was done using Hidden Markov Models and other statistical methods [7]. Audio signals can be directly embedded into vector form similar to the way words are embedded into vector form for natural language processing. This opens up future scope for this recommendation research where audio signals can directly be used to make vector representations instead of fetching textual data based on the audio and converting it to vector representations.

The initial input in this case will be raw audio spoken out by the user. The model will take in the raw speech audio and process it to convert it to a text output for the latter stages of the system.

The audio samples used are 2 seconds long. They are sampled at 44.1KHz and are dual-channel audio signals. As a part of preprocessing, the author converts audio files into NumPy arrays and saves them separately for further processing. It consists of the following steps:

1. Read the audio file and compute the MFCC using librosa library.
2. MFCC vectors might vary in size for different audio inputs.
3. Zero paddings is applied to all MFCC vectors to make them of uniform size.
4. These are then stored as NumPy array files.

The training labels are one-hot encoded so that there is no bias by virtue of the name of the class.

The model has been modified to suit the project's needs of accurate hearing of both song and artist name so as to recommend similar songs. After testing for different values and noting down the accuracies, the architecture consists of the following layers (in order):

1. Conv2D Layer with kernel size 2x2
2. Max Pooling with size 2x2
3. Dropout of 25%
4. Flattening Layer
5. Fully Connected Layer
6. Dropout of 50%
7. Dense Softmax Output Layer

The accuracy of the model is compared on two datasets. The first one is the Tensorflow Speech Command Dataset by Google consisting of over 105,000 WAVE audio files of people speaking out 30 different words [8]. The accuracy for the training and testing set was over 90%.

At the end of this step, the audio is converted into text that can be used for further processing as explained in detail in the further sections, in order to find a vector representation of the input song's lyrics and compare it against all the other song lyrics representations to determine similar songs.

B. Doc2Vec Embeddings Model

Word2Vec is one of the most efficient models to learn word embeddings using shallow neural networks [9]. Word embeddings are representations of the total document vocabulary and help in gaining context of words and documents, while also helping to understand the relationship between different words and documents. Word2Vec can be extended to Doc2Vec where instead of embedding just words, entire documents are analyzed in the form of vector representations that can be compared [10]. Thus, Doc2Vec in the context of music recommendation can help find similar lyrics based on a given set of lyrics.

To train the model, a large dataset of music lyrics is needed. For this purpose, the musiXmatch dataset is used. The musiXmatch dataset consists of all the lyrics of the most widely used Million song dataset [11]. It consists of a total of 2,37,662 song lyrics that can be used to efficiently train any music model.

Owing to copyright issues, these lyrics for songs are given in a bag-of-words format. Bag-of-words indicates that the words of each song are jumbled together and not presented in a linear manner [12]. Since the end goal is to use a Doc2Vec embeddings model that uses vector representations, this bag-of-words format is not that big an issue.

Once all the lyrics have been extracted from the dataset, the next step is to form high-dimensional embedding vectors for these lyrics so as to make the model for the recommendation. This can be done through a technique called Distributed Bag of Words Paragraph Vector (PV-DBOW) model [13].

A PV-DBOW model works by picking a paragraph matrix in each epoch or iteration and words from the chosen paragraph are picked randomly and sampled in an attempt to predict the word with the input being the paragraph matrix. This way, the model weights are updated based on the accuracy of each prediction of each iteration. In simpler terms, a PV-DBOW model can be viewed as a paragraph version of a skip-gram model of Word2Vec.

In the case of this paper, each paragraph is a specific song's lyrics. Implementing this model will result in all songs being mapped in high-dimensional vector representations that can now be manipulated and used to calculate the similarities between songs through the lyrics.

Additionally, in order to visualize the high dimensional data of the model for a better level understanding of how the vectors are represented, a dimensionality reduction technique such as at-distributed Stochastic Neighbour Embedding (t-SNE) can be used [14]. This technique uses the neighborhoods of all the vector points in the high dimensional space where the lyrics vectors are represented and by estimating a probability distribution, manipulates the distribution in a lower-dimensional space so that the similarities between songs can be viewed in a matrix or graphical format.

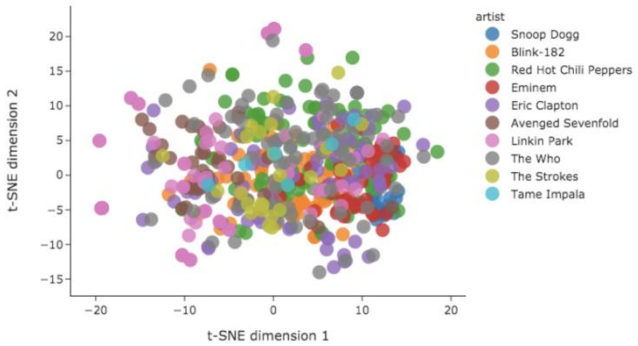

Fig 1: t-SNE representation of songs of 10 artists

Fig 1 represents a low-dimensional view of songs by ten different artists while preserving the relationship maintained between them in the original higher dimension. Doing so gives a direct view of what songs are closely related by lyrics are what is not at all. For example in the above graph, most of the brown points and red points are not related and are mostly at the two ends of the graph, meaning that songs by the brown artist Avenged Sevenfold are very different from the songs by the red artist Eminem, as they follow two very different genres of Rock and Rap and have very different styles of music. Thus, by such a representation, a clearer idea of how songs are related to each other can be conveniently visualized.

Thus, once the model is ready and established, song recommendations can be made using the concept of Cosine Similarity. Cosine Similarity is the most popular technique to find the distance between two vectors in all Word2Vec and Doc2Vec models [15]. Taking two vectors in the inner product space, cosine similarity measures the similarity, i.e the cosine of the angle between the two-song vectors in this case and determines if they roughly point to the same direction.
The vector representations will be similar to what is seen in Fig 1 but at a higher dimension. The distance and direction vectors in the higher dimension are used to find the cosine similarity and the songs that are clustered together and have similar direction vectors will end up being similar, which is what is represented in Fig 1 at a reduced dimension for better understanding of how the process occurs.

Thus, given an input song, the vector space of the model is searched, and using cosine similarity, the similarity of this input song is calculated with all other songs, and similar songs are outputted. This way, given a song, we can use a Speech Recognition system and a Doc2Vec embeddings model to find similar songs based on the lyrics through vector representations and cosine similarity.

IV. Limitations And Future Scope

The paper efficiently recommends songs based on lyrics and works flawlessly on all systems. However, instead of recommending songs based on just one song, finding similar songs based off an entire list of songs will be more efficient. This way, the model has 10-20 different songs to work with while trying to find other similar songs, naturally being better.

Similarly, using just lyrics will give similar songs but adding the sound of music as an input feature along with the lyrics will help in finding similar songs that are otherwise tricky to pair together. This will definitely make the user experience better.

The paper aims to build on these features in the future and scale the system further for better performance and user experience.

V. Conclusion

Content-based filtering and recommendation systems in the field of music have been here for a long time. However, as technology advances, the methods of recommendation improve and open up new avenues and possibilities for better performance and a more friendly user experience.

The aim of this paper is to ease the process of finding songs. If a person simply speaks out a song for which they want similar recommendations, the model used in the paper will immediately provide similar songs with listening options so that people can easily access similar songs based on any song's lyrics.

References

[1] Wang, D., Liang, Y., Xu, D., Feng, X., & Guan, R. (2018). A content-based recommender system for computer science publications. *Knowledge-Based Systems*, *157*, 1-9.
[2] He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T. S. (2017, April). Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web* (pp. 173-182).
[3] Fessahaye, F., Perez, L., Zhan, T., Zhang, R., Fossier, C., Markarian, R., ... & Oh, P. (2019, January). T-recsys: A novel music recommendation system using deep learning. In *2019 IEEE International Conference on Consumer Electronics (ICCE)* (pp. 1-6). IEEE.
[4] Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., ... & Ng, A. Y. (2014). Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*.

[5] Kim, D., Seo, D., Cho, S., & Kang, P. (2019). Multi-co-training for document classification using various document representations: TF–IDF, LDA, and Doc2Vec. *Information Sciences*, *477*, 15-29.
[6] Luo, C., Zhan, J., Xue, X., Wang, L., Ren, R., & Yang, Q. (2018, October). Cosine normalization: Using cosine similarity instead of dot product in neural networks. In *International Conference on Artificial Neural Networks* (pp. 382-391). Springer, Cham.
[7] Eddy, S. R. (2004). What is a hidden Markov model?. *Nature biotechnology*, *22*(10), 1315-1316.
[8] https://www.tensorflow.org/datasets/catalog/speech_commands
[9] Church, K. W. (2017). Word2Vec. *Natural Language Engineering*, *23*(1), 155-162.
[10] Lau, J. H., & Baldwin, T. (2016). An empirical evaluation of doc2vec with practical insights into document embedding generation. *arXiv preprint arXiv:1607.05368*.
[11] http://millionsongdataset.com/musixmatch/
[12] Zhang, Y., Jin, R., & Zhou, Z. H. (2010). Understanding bag-of-words model: a statistical framework. *International Journal of Machine Learning and Cybernetics*, *1*(1-4), 43-52.
[13] Mani, K., Verma, I., Meisheri, H., & Dey, L. (2018, December). Multi-document summarization using distributed bag-of-words model. In *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)* (pp. 672-675). IEEE.
[14] Wattenberg, M., Viégas, F., & Johnson, I. (2016). How to use t-SNE effectively. *Distill*, *1*(10), e2.
[15] Lau, J. H., & Baldwin, T. (2016). An empirical evaluation of doc2vec with practical insights into document embedding generation. *arXiv preprint arXiv:1607.05368*.