Deep Rapping: Character Level Neural Models for Automated Rap Lyrics Composition

Aaron Carl T. Fernandez, Ken Jon M. Tarnate, Madhavi Devaraj

Abstract: "Dope", "Twerk", "YOLO", these are just some of the words that originated from rap music which made into the Oxford dictionary. Rap lyrics break the traditional structure of English, making use of shorten and invented words to create rhythmic lines and inject informality, humor, and attitude in the music. In this paper, we attack this domain on a computational perspective, by implementing deep learning models that could forge rap lyrics through unsupervised character prediction. Our work employed novel recurrent neural networks for the task at hand and showed that these can emulate human creativity in rap lyrics composition based on qualitative analysis, rhyme density score, and Turing test performed on computer science students.

Keywords: Gated Recurrent Unit; Long Short-Term Memory; Natural Language Generation; Recurrent Neural Networks.

I. INTRODUCTION

Rap lyrics are stigmatized as offensive, profane, and inappropriate, heavily laced with words that belong to the semantic field of sex, drugs, and discrimination [1]. To justify our inclination on such domain for scholastic reasons, William Shakespeare, whom has been credited by the Oxford English Dictionary for coining up to 2,000 words, is found to be out worded by most modern-day rappers such as Aesop Rock, who has 7,392 unique words under his belt compared to the celebrated poet who only had 5,170 [2].

This insinuates that rap lyrics have larger vocabulary than the English literature and opens the question "How can we build an artificial intelligence that can come up with a vocabulary of such magnitude?". The question led us to deep learning, a subset of artificial intelligence that can generate music, text and motion capture [3] using multi-layered artificial neural networks. Artificial neural networks, specifically recurrent neural networks, have an internal memory that maintains previously calculated results, allowing information to persist. There is an abundance of its application in the current literature, along with its successors, Long Short-Term Memory and Gated Recurrent Unit. This paper investigates the automatic composition of rap lyrics on different recurrent neural network architectures and assess its quality through cross-entropy loss, rhyme density score and human evaluation. Rhyme is considered as an important characteristic feature of rap lyrics [4] but in this study, we put more weight on the merit of how close machines emulate human writing.

II. LITERATURE REVIEW

This work has been inspired by DeepBeat [5], an online rap lyrics generator tool built on the RankSVM algorithm and multi-layered artificial neural networks. It is a prediction model that assembles rap lyrics line-by-line using intact lines from existing rap songs. The results were promising as it shows that it can outperform top human rappers in terms of length and rhyme frequency by 21%.However, we differentiate this work by developing an artificial intelligence, which not just intertwines existing rap lines to generate lyrics but instead, build this on the character level that could resemble human performance as closely as possible.Syntactic text generation is further broken down to sentence-level, word-level and character-level. The latter is exemplified in the generative language model of Sutskever et al. [6], wherein the authors trained a recurrent neural network on the character sequences of Wikipedia, New York Times and machine learning papers from Conference on Neural Information Processing Systems and Journal of Machine Learning Research. It was considered as the largest implementation of recurrent neural network at that time, training their model of 500 hidden layers with 1,500 units each for five days on a parallel system of 8 high-end GPUs with 4GB RAM each and data amounting up to 300MB.

Their work has proven that a large vocabulary of words, grammatical and punctuation rules can be learned at the character level. This has further been bolstered by Graves [3], which achieved the same success on a relatively smaller data and with a different type of recurrent neural network called the "Long Short-Term Memory" neural network or LSTM [7], a type of recurrent neural network, which can amend its predecessor's instability when generating sequences. His work has shown that character-level LSTMs can outperform word-level LSTMs on discrete sequence generation and argued that predicting one character at a time allows a more interesting text generation as it is capable of inventing novel words and strings [3].Having said that, writing rap lyrics using "*Long Short-Term Memory*" neural networks had already been executed by Potash et al. [8], whose objective is to generate rap lyrics in the similar style of a given rapper but not identical to his existing lyrics, emulating the task of ghostwriting. They compared their work with the model of Barbieri et al. [9], which employed constrained Markov processes on the same task.

Revised Manuscript Received on December 28, 2018
Aaron Carl T. Fernandez Mapua University, Manila, Philippines
Ken Jon M. Tarnate Mapua University, Manila, Philippines
Dr. Madhavi Devaraj Mapua University, Manila, Philippines

To quantify the critical aspect of their experiment, which is to produce similar yet different lyrics, they evaluated their models by correlating the cosine similarity between the existing and generated lyrics using the "*Inverse Document Frequency*" algorithm, as well as, computing its "*Rhyme* Density" score.

Rhyme density is the total number of rhymed syllables over the total number of syllables. It was formulated by Hirjee et al. [10], who defined different methods of racking up potential rhymes based on the phonetic frequencies in rap lyrics. It is worth noting that it has not only been applied in empirical works in rap music [5, 8, 11] but in lyrical analysis [12], and computational poetry evaluation [13] as well.While the authors had been successful in exhibiting how LSTMs outperformed their baseline Markov model in producing rap lyrics that has the same rhyming style of the target artists, a more recent recurrent neural network called "*Gated Recurrent Unit*" [14] has demonstrated to converge faster and perform better than LSTMs on modelling tasks such as polyphonic music and speech signal generation [15]. Although, it has been noted that the results were still preliminary and not conclusive. Prospective experiments on these contemporary gated units is a viable supplement in its current literature.

III. METHODOLOGY

3.1. Character Level Language Model

Generating text one character at a time often yield worse results compared to word–level language models [3]. The latter can make decisions at a coarser granularity and has a lower probability of producing spelling errors, which the former suffers at. However, rap lyrics often put into use large amounts of colloquial vocabulary and slangs from the subculture [16] such as "*fam*", "*shawty*", "*gangsta*", "*po-po*", "*OG*" etc. Moreover, real-world rap lyrics available in the internet are usually crowdsourced, making it susceptible to inconsistencies, typographical errors, and spelling mistakes. This can significantly increase the size of the vocabulary by having several versions of the same word [17] resulting to extremely large output nodes prone to floating-point errors and underflow that could severely deteriorate the quality of its predictions. In addition, novel encoding techniques of input words in a word-level language model may not be able to capture the similarity of the written form of the words which could result to poor representation of infrequent words in the training data [17]. Hence, we opt to model our rap lyrics at the level of characters to overcome these limitations. Our language model first, finds some good default seed strings in the training corpus. This is done heuristically by splitting the data per new line and creating a random sample of 200 unique lines from it, after which, we only take the top quartile of this random sample based on each line's length for the reason of viewing longer lines as a more decent seed string. Then, each character in the training corpus is tokenized, preserving the original letter case of the letter tokens and sorting it in a descending order according to how common it appears. The seed strings and character tokens are both preserved as meta-models, to be used during the rap lyrics generation. Then, the training data is converted into a vector of integer numbers using the character tokens. The resulting data vector is reformatted into input and target sequences, which what will be fed into the recurrent neural network. The target sequences are configured to have only one time-step ahead of the input sequences to force the algorithm to make predictions one character at a time.The reformatting starts by taking strips of the input and target data vectors at an interval of 25 characters and cutting the resulting strips into 50-character sequences after which, the resulting sequences are stacked altogether. The resulting data is then again reformatted so that the first sequence in the n^{th} batch picks up exactly where the sequence in the $(n-1)^{th}$ batch left off because a recurrent neural network cell state does not reset between batches in a "*stateful*" model. Finally, the target data is embedded with an extra axis to work with the sparse categorical cross-entropy loss function, which is what will be used to evaluate the training of the recurrent neural network. Once the neural network is fully trained, an initial seed is picked out randomly and transformed into a vector of integer numbers using the preserved meta-model earlier. The succeeding character tokens with the maximum probability are then predicted and appended into the vector one at a time. The character token prediction employs a temperature parameter, which value is used to divide the natural logarithm of the probabilities array to smoothen or sharpen it. The intuition behind this is that a lower temperature value dictates a more conservative rap lyrics generation while a higher value will be more creative and diverse at the cost of more mistakes. The prediction iterates depending on the length of the rap lyrics to be generated, which is specified by the user. Once finished, the resulting integer vector is de-tokenized, forging a new set of rap lyrics.

3.2 Evaluation Metrics

Sparse Categorical Cross Entropy

Guided by the Shannon's source coding theorem [18], the minimum space we can compress the information in song lyrics is given by the entropy of the distribution over the lyrics space. This is cannot be computed directly. As an alternative, its approximation by using the language model to predict a set of lyrics it has not seen before can be used. This approximation is called the cross-entropy measure [19]. Specifically, the sparse categorical cross-entropy, which is a multiclass logarithmic loss that measures the dissimilarity between the target distribution y and the predicted distribution $y^{\sim}$ formulated as:

$$L_{cross\ entropy}\ (y^{\sim}, y) = -\Sigma_i y_i \log(y_i^{\sim}) \quad (1)$$

Rhyme Density

The rhyme density measure has been introduced in [10] as a quantitative measure of the technical quality of rap lyrics from a rhyming perspective. It is formulated as the average length of the longest rhyme per word [10] and is the same metric used to evaluate the rap lyrics generation models of [5] and [8]. A tool called the "*Rhyme Analyzer*" [20] which calculates the statistical features of the rhymes detected in

the input lyrics has been developed by the same authors of [10] and is available for download[1].

Turing Test

Finding human evaluators who are well versed in the technicality of rap lyrics can be challenging [5, 21]. But following Alan Turing's paper "Computing Machinery and Intelligence", wherein a machine is proposed to be tested based on how closely it can resemble human's intelligent behavior [22], enables us to attain the objective truth on the quality of our rap lyrics by questioning human evaluators abstractedly if it was written by a human or a machine. Since the primary objective of this experiment is to examine if a recurrent neural network can produce rap lyrics at the character level that are admissible to humans, we sampled our lyrics to 30 computer science students of Mapua University, of whom 2 were at the doctorate level, 7 were at the master level, and 21 were undergraduates. All participants were informed that the test was for a machine-learning project to get a more conscientious evaluation. We also asked the participants if they are familiar with rap lyrics to which 13 responded "yes" and 17 responded "no". For this test, we sampled 10 6-bar verses for each of the RNN model, on both datasets, and for temperatures 0.2, 0.5, and 1.0 to test samples of low, medium, and high diversity respectively. Of the 10 samples generated for each criterion set, we selected the best sample to be included in the test in terms of least spelling mistakes and subjective quality. The participants were given three sets of tests, which were for low, medium, and high diversity set of generated lyrics. Each set has eight 6-bar rap lyrics, of which six were generated from our models, one generated from deepbeat.org, and a human-written rap lyric taken from ohhla.com. The participants were asked to make a binary decision (human or machine) about the source of the rap lyrics, which scores 1 point for each lyric perceived as "human" by the participants.

IV. EXPERIMENTS

4.1. Data

We collected 4,799 rap lyrics from The Original Hip-Hop (Rap) Lyrics Archive[2], the same lyrics source of [8]. The corpus was cleaned by stripping all HTML tags, metadata (*artist name, song name, album name, INTRO, CHORUS, Repeat 2x, 4x, etc*.), and whitespaces between lines. We also manually normalized the data to filter out any malformed metadata and non-English songs, which yielded 696,787 lyric lines amounting to a 27.1MB file. We refer to this corpus as $Dataset_{large}$.

We hypothesized that the vast lyrical diversity among the rap lyrics collected may degrade the quality of the generated rap lyrics in terms of rhyme density. We are also interested to examine the performance of the recurrent neural network architectures discussed above on a smaller dataset, and see the disparity compared to a larger dataset. Hence, we created another corpus concentrated on the songs of three rap artists namely: Notorious B.I.G, Fabolous, and Lil' Wayne, as these are the rappers identified in [8, 26], who attained the best rhyme scores. This brought in 42,918 lyric lines in a 1.8MB file, which we refer to as $Dataset_{small}$.

4.2. Training Hyperparameters

We tried different hyper-parameter settings on a plain recurrent neural network and using $Dataset_{large}$, and found that three hidden layers with 512 neurons each, optimized by ADAM [23] with a learning rate of 0.0001, may work best for the task at hand. We applied the same hyper-parameter settings on all the recurrent neural network models and on both $Dataset_{large}$ and $Dataset_{small}$ to get a fair judgement on their performances for this experiment. We trained all models on a 12GB Tesla K80 for 400 epochs but with an early stopping rule if there is no loss improvement for 5 epochs on $Dataset_{small}$ and 3 epochs on $Dataset_{large}$.

V. RESULTS AND DISCUSSION

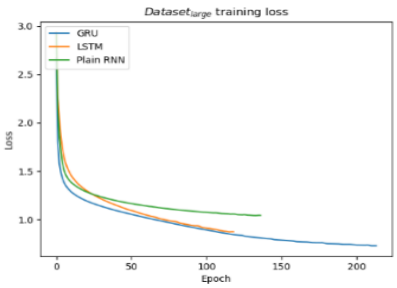
5.1. Training Results

Table 1 presents the actual training time clocked by our RNN models on both datasets. Clearly, plain recurrent neural networks converged the fastest, taking only almost 2 hours to train on a 1.8MB dataset and 15 hours on a larger dataset of 27.1MB. Having said that, it was the worst performer on both datasets in terms of cross-entropy loss.

The disparity in the runtimes of LSTM and GRU on both datasets suggests that GRU trains faster than LSTM on a small dataset but can be outran by the latter on a larger dataset. However, this claim is still refutable, as a more thorough experiment concentrated on this facet is required to make such conclusive assertion. In terms training quality, the shape of the training curves in Figure 1 indicate that the learning rate employed was just about right on all RNN models but scaling it down a little could have improved the training. GRU outperformed LSTM and plain RNN on both datasets. Although, the performance of LSTM is not far off GRU.

Table 1. Clocked run times of plain recurrent neural network, long short-term memory, and gated recurrent unit on both datasets.

	PRNN	LSTM	GRU
$Dataset_L$	00:14:57:32	02:04:08:07	03:01:02:56
$Dataset_S$	00:01:39:09	00:08:06:02	00:06:14:01



[1] https://sourceforge.net/projects/rhymeanalyzer/

[2] http://ohhla.com/

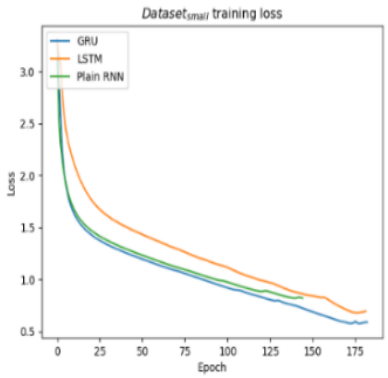


Figure 1. Training loss of plain recurrent neural network, long short-term memory, and gated recurrent unit on $Dataset_{large}$ (left) and $Dataset_{small}$ (right).

5.2. Qualitative Analysis

Table 2 exhibits the generated rap lyrics when the temperature value was set to 0.5. All models learned how to capitalize proper nouns and starting letters of each verse. We manually verified that all verse generated are original and entirely invented by our model, except for the first lines, which were the seeds that were taken from our datasets.

It is hard to judge the coherence of the generated rap lyrics due to the vast grammatical freedom in rap music. This makes it more difficult to differentiate the lyrical quality of each model subjectively as its distinctiveness is imperceptible except for the minor spelling mistakes in the plain RNN samples on both datasets such as "*connerfuckin*", "*heacher*", and "*eeling*", which do not even exist in our datasets. Although, this reinforces Grave's statement in [3], which mentioned that character level language models are capable of inventing new words on their own.

For the above reasons, we obtained an evaluation on an unbiased perspective such as rhyme density calculation and Turing test to get a more reliable assessment. These are discussed on the succeeding sections.

Table 2. Generated Rap Lyrics Sample of Medium Diversity (Warning: Explicit Content).

Generated Lyrics	Model
Only 2 years old when daddy used to bring them I don't know she me now cut I'm a flavorin' It hurtin' and got a gun cold car anything I got that blow blow, on ya mouth like man I'm so steel I'm a motherfuckin Cash Money Millionaire I'm a move ass connerfuckin niggas start shooking	PRNN D_S
I been peeping you too, nigga I see you shining We ride this singles, hit the parkin' shit and then I'm talkin' 'bout who, it's no fuckin' in my sky is Stop playin' what try to fly the biggy down the faster The niggas throwin splats, around the grave She tell me what they be something	LSTM D_S
But what did you say, you said that you never change You can call me When I'ma Holly Grove I know my homies from Hollywood, this ain't pointin I ain't fuckin' with the bullet that's cool, not too dead I say fuck nigga pull up, I could use a baby Girl I did it	GRU D_S
And I got like ten bitches in the car with me there with that pussy I'm a heacher day in the hood now I said I was a youngin' tell 'em I'm straight up I got a black bags on the beat like a salad blade I still really get to see a bitch and a nigga eeling in the sky I can't trust a nigga out you	PRNN D_L
Now your kids gotta deal with this shit cause you do We goin' out to the moon and we get to the hood Cookin' gin, continue to me It's like the moon months saying to the people that they say What's the deal with these hoes, and they want to see me Acting all me about where they home nigga	LSTM D_L
You stuck dat kush inside dem sickles lot we got high And we gon' ride on you can see We gone my niggas gon bang And we gon' be gettin' outta line I know that you're lookin' like I ain't got this shit though I ain't got no signs to the note	GRU D_L

5.3. Rhyme Density Scores

We generated 100 16-bar verses of medium diversity for each of the RNN models and on both datasets like in [5]. We considered one bar as equivalent to one line following [24], which mentioned that a typical 16-bar verse in a rap song is composed of 16 lines. We used the "Rhyme Analyzer" tool [20] developed by Hirjee et al. [10] to analyze the rhyme densities of our generated rap lyrics. This is presented on Table 3 and as shown, our generated rap lyrics suffered from poor rhyme density score, which is distant to what DeepBeat had achieved and did not even reach the threshold of human rappers such as MC Hammer and Ice Cube, who had the lowest rhyme density of 0.19 [10]. This is expected as we did not consider rhymes programmatically. We only hypothesized that our generated rap lyrics would inherit the rhyme densities of our corpus, which were 0.28 for $Dataset_{large}$, and 0.32 for $Dataset_{small}$.

Table 3. Average Rhyme Densities of all Trained Models.

Model and Dataset	Rhyme Density Score
PRNN D_S	0.14
LSTM D_S	0.13
GRU D_S	0.12
PRNN D_L	0.15
LSTM D_L	0.13
GRU D_L	0.15

5.4. Turing Test Results

Figure 2 shows that DeepBeat's generated rap lyrics attained the highest score, being perceived as human-written by 71% of the total participants not familiar with rap lyrics, even outperforming the real human-written lyrics by one vote. This was almost matched by the LSTM model trained on the Dataset_small, which achieved the best human evaluation out of all our trained models, deceiving 67% and 53% of the participants whom are acquainted and unaccustomed to rap lyrics respectively. The generated rap lyrics of plain RNN and GRU on both datasets received satisfactory evaluation, deceiving half of the participants on all tests. Overall, the result is positive as all our generated rap lyrics deceived a significant number of participants into thinking that it was written by human rappers, which is the primary goal of this experiment. We can attribute this success to the offbeat structure in rap lyrics and its unorthodox vocabulary. This worked in favor of character level text generation as mistakes in spelling and grammar, and even its incoherence can be confused with its whimsical nature. Having said that, generating lyrics for other genres, which take form in lyrical coherence and conventional vocabulary, may be difficult to generate at the character level. Hence, generation at a higher level such as word, phrase, and sentence would be more appropriate.

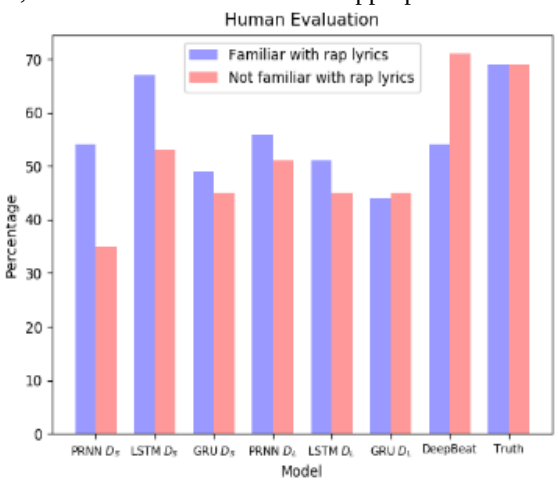


Figure 2. Human Evaluation in Percentage.

VI. CONCLUSION

We have investigated the performance of plain recurrent neural network, long short-term memory, and gated recurrent unit on the domain of automated rap lyrics composition at the character level on small and medium-sized dataset. Plain recurrent neural network converged the fastest but obtained the worst cross-entropy loss. GRU outperformed both plain RNN and LSTM in terms of training quality but was the slowest to train on a medium-sized dataset, taking more than 3 days to converge. Having said that, it trained 25% faster than LSTM on a smaller dataset. This suggests that the convergence of LSTM and GRU may depend on the dataset size. However, this assertion demands a more thorough and concentrated investigation. All RNN models on both datasets learned the basic syntax sentence structure of a human-written rap lyric, even when built at the character level. Our machine-generated rap lyrics convinced a significant number of computer science students into thinking that that it was written by a human rapper based on the Turing test performed. Although, it suffered low rhyme density as it was not considered programmatically. Finally, a potential extension of this work is the incorporation of rhymes and intelligibility in the algorithm to generate more rhythmic and coherent rap lyrics. A hypothetical solution is to generate rap lines at the character level and develop an algorithm that will weave the generated rap lines into a rap verse based on its accidence and rhyme density score. The insistence of initially constructing the rap lyrics at the character level is due to its offbeat structure and unorthodox vocabulary, which is assumed difficult if built on a word level.

REFERENCES

1. M. Escoto and M. B. Torrens. Rap the Language. Publicaciones Didacticas 28 **(2012)**
2. M. Daniels. The Largest Vocabulary in Hip Hop. Retrieved August 5, 2018 from https://pudding.cool/2017/02/ vocabulary/
3. Graves. Generating Sequences with Recurrent Neural Networks. arXiv:1308.0850, **(2013)**
4. K. Addanki and D. Wu. Unsupervised rhyme scheme identification in hip hop lyrics using hidden markov models. Proceedings of the First international conference on Statistical Language and Speech Processing, **(2013)**
5. E. Malmi, P. Takala, H. Toivonen, T. Raiko, and A. Gionis. DopeLearning: A Computational Approach to Rap Lyrics Generation. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, **(2016)**
6. Sutskever, J. Martens, and G. Hinton. Generating text with recurrent neural networks. Proceedings of the 28th International Conference on International Conference on Machine Learning **(2011)**
7. S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. Neural Comput. 9, 8 **(1997)**
8. P. Potash, A. Romanov, and A. Rumshisky. GhostWriter: Using an LSTM for Automatic Rap Lyric Generation. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing **(2015)**
9. G. Barbieri, F. Pachet, P. Roy, and M. D. Esposti. Markov constraints for generating lyrics with style. Proceedings of the 20th European Conference on Artificial Intelligence **(2012)**
10. H. Hirjee and D. Brown. Using Automated Rhyme Detection to Characterize Rhyming Style in Rap Music. Empirical Musicology Review 5, 4 **(2010)**
11. N. Condit-Schultz. MCFlow: A Digital Corpus of Rap Transcriptions. Empirical Musicology Review 11, 2 **(2017)**
12. M. Fell and C. Sporleder. Lyrics-based Analysis and Classification of Music. Proceedings of 25th International Conference on Computational Linguistics **(2014)**
13. E. Lamb, D. G. Brown, and C. Clarke. Can Human Assistance Improve a Computational Poet? Proceedings of Bridges 2015: Mathematics, Music, Art, Architecture, Culture **(2015)**
14. Cho, B. Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing **(2014)**
15. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. NIPS 2014 Workshop on Deep Learning **(2014)**
16. Wu, K. Addanki, and M. Saers. Freestyle: A Challenge-Response System for Hip Hop Lyrics via Unsupervised Induction of Stochastic Transduction Grammars. Proceedings of the Annual Conference of the International Speech Communication Association **(2013)**
17. P. Bojanowski, A. Joulin, and T. Mikolov. Alternative structures for character-level RNNs. arXiv:1511.06303 **(2016)**
18. E. Shannon. A mathematical theory of communication. SIGMOBILE Mob. Comput. Commun. Rev. 5, 1 **(2001)**

19. P. T. de Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein. A Tutorial Introduction to the Cross-Entropy Method. Annals of Operations Research 134, 1 **(2005)**
20. H. Hirjee and D. G. Brown. Rhyme Analyzer: An Analysis Tool for Rap Lyrics. Proceedings of 11th International Society for Music Information Retrieval Conference **(2010)**
21. Wu, K. Addanki, M. Saers, and M. Beloucif. Learning to Freestyle: Hip Hop Challenge-Response Induction via Transduction Rule Segmentation. Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing **(2013)**
22. M. Turing. Computing machinery and intelligence. In Computers & thought **(1995)**
23. D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. Proceedings of the 3rd International Conference on Learning Representations **(2015)**
24. Paul Edwards. How to Rap: The Art & Science of the Hip-Hop MC, Chicago: Chicago Review Press **(2009)**

AUTHORS

Aaron Carl T. Fernandez
He obtained his bachelor's degree in information technology from San Beda College – Alabang, Philippines, in 2011 and is currently working toward a M.S. degree in computer science at Mapua University, Philippines. He is also a full time Technology Lead specializing in mainframe applications development for Infosys. His current research interests are sequence modelling using artificial neural networks and deep reinforcement learning for playing games.

Ken Jon M. Tarnate
He obtained his bachelor's degree in information and communication technology education from Philippine Normal University – Manila, Philippines, in 2014. He is currently working toward a M.S. degree in computer science at Mapua University, Philippines. He is also a full-time scholar of Engineering Research and Development for Technology (ERDT) under Department of Science and Technology of the Philippines.

Dr. Madhavi Devaraj
She graduated with a PhD in Computer Science from Dr. A.P.J. Abdul Kalam Technical University (formerly Uttar Pradesh Technical University) in Lucknow, Uttar Pradesh, India in 2016. She took up her Master in Philosophy in Computer Science from the Madurai Kamaraj University and Master of Computer Applications from V.V.Vannaiaperumal College for Women both in India in 2004 and 2000 respectively. She finished her Bachelor of Science in Mathematics from the Bharathidasan University - Government Arts College for Women in 1997.