Creating and Evaluating a Lyrics Generator Specialized in Rap Lyrics with a High Rhyme Density

MATTHIAS WENTINK, University of Twente, The Netherlands

Current musical tools and education allow every day people to advance their musical skills, however, they do not offer creative stimulation. One aspect in which creative stimulation could be applied within music is lyrics. This research proposes an architecture for a rap lyrics generator that focuses on multisyllabic rhyme; rhyming on more than one syllable that can span multiple words. This is done by selecting rhyming words from an input sentence and filling the space between them with Bidirectional Encoder Representations from Transformers (BERT). A prototype for this architecture has been developed and evaluated which showed that generated lyrics score significantly lower than existing human-written lyrics on rhyme and coherence. However, the participants are, at times, also impressed by the output of the tool. This suggests that an improved implementation of the proposed program could be viable for generating rap lyrics that contain multisyllabic rhyme and coherence at a higher level.

Additional Key Words and Phrases: Computational Creativity, Lyrics Generation, Multisyllabic Rhyme, BERT

1 INTRODUCTION

Computational creativity is a subtopic of Artificial Intelligence (AI). It is about using AI technology to make works that can be considered creative. Applications can be in any art or creative field; this paper focuses on the field of music. More specifically, it uses Computational Creativity to help people compose lyrics for rap music.

Nowadays, independent musicians are becoming increasingly more prevalent. These musicians create and publish music with very limited musical facilities. Instead of using instruments and studios, they heavily rely on powerful music software and virtual instruments to produce their songs[18]. When the songs are finished, they use social media to spread their music instead of through a record label [18]. There are a lot of modern tools that enable a person to become a successful musician, however these tools lack creative stimulation. Very little musicians have the creative capacity to do everything that they want within their music genre. For example, producers that can make beats but who find it difficult to write lyrics. Computational Creativity can provide creative stimulation by generating suggestions on the part of music creation that a user finds difficult. This can help musicians that are less well rounded in independently creating music.

Lyrics are very important to many forms of music and can also be generated by an AI. Rhyme is crucial to lyrics because it gives structure [7] and this makes a song recognizable and catchy for the listeners. In rap music, rhyme is especially important as it helps in delivering interesting rhythms and it shows off the skills of a rapper[7]. However, coming up with a coherent story that rhymes is a hard and creative task.

> Living off borrowed time, the clock **ticks faster**.
> That'd be the hour they knock the **slick blaster**.
> **Dick Dastar**dly and *Muttley* with **sick laughter**.
> A gunfight and they come to *cut the* **mixmaster**

Lyrics 1. Four lines from the song "Accordion" by Madvillain[17]. Rhyming words have been highlighted by using the same text decoration.

If advanced rhyming methods of Hip-Hop, as can be seen in lyrics 1, are taken into account, writing lyrics becomes more complex. A rhyme technique that is often used in rap music is multisyllabic rhyme; rhyming more than one syllable in words or a group of words. They can be perfect rhymes (e.g. "slick" and "sick"), but imperfect rhymes are more often used (e.g. "faster" and "laughter"). Furthermore, multisyllabic rhymes can be internal rhymes; meaning that the rhyme does not occur at the ends of sentences but occurs within a sentence. This is illustrated in the example with "Dick Dastardly" and "sick laughter".

Using AI to combat the creative task of creating coherent rap lyrics containing multisyllabic rhyme can help lyricists in creating better music. In doing so, they can possibly find a creative outlet. To provide this, this research serves the following two goals:

(1) To create an application that, given an input sentence, can create a sentence where multiple syllables rhyme with the input, while maintaining a logical sentence structure.
(2) To evaluate the lyrics generated by the application based on rhyme quality and coherence of the input and output lines.

To achieve the mentioned goals, the following main research question is established:

How can an application be designed that generates a sentence that has a high rhyme density with a given input sentence, while maintaining coherence with the input sentence?

This question can be split up into the following two sub research questions:

(1) How can an application be made that generates a coherent rhyming sentence to an input sentence?
(2) To what extent can such an application support an artist in generating good rhymes and coherent lyrics?

TScIT 38, February 3, 2023, Enschede, The Netherlands

2 RELATED WORK

Firstly, it is discussed how rhyming works and can be computationally understood. Secondly, existing rap lyrics generators are discussed and assessed on whether their features can be used for this research.

2.1 Rhyming

It is important to note that spelling of words has very little to do with whether two words rhyme in the English language. For example, "pour" and "hour" do not rhyme whilst their spelling is similar, and "break" and "shake" do rhyme whilst their spelling is different. Instead of spelling, the pronunciation of words should be looked at to detect rhyme.

Rhyme between two words takes place between the vowels and the consonants after the vowels of the syllables. It is called a perfect rhyme when the vowels and end consonants sound identical, and an imperfect rhyme when either the vowels or end consonants are not a perfect match. For imperfect rhymes, the phonemes that do not match perfectly should still bear some resemblance (e.g. "M" and "N"). Furthermore, syllables also rhyme better when they have a matched strong stress level. Rhyme can take place in different places within a verse. When rhyme is present at the end of lines, it is called end rhyme. When it occurs within a line, it is called internal rhyme.

To let an AI identify different types of rhyme, words should be converted into their pronunciation. The Carnegie Mellon University (CMU) pronunciation dictionary[9] is often used for denoting the pronunciation of words in ARPABet. ARPABet is a set of transcriptions for phonemes created by the Defense Advanced Research Projects Agency (DARPA). The vowels are also denoted with a stress score of 1 for primary stress, 2 for secondary stress and 0 for no stress. With this dictionary of phonemes, it becomes possible for a computer to understand perfect rhyme. For example, if you take the words "break" and "cake", which are perfect rhymes but with very different spellings, it will be split up as B R EY K and K EY K. Since the last two phonemes (EY and K) match between the two, it is evident that it rhymes. To extend this technique for imperfect rhymes, a score is needed to determine how well phonemes rhyme with each other. For example, the words "frame" and "rain" will be split up as F R EY M and R EY N. Here, "M" and "N" are not identical but, with scores for phoneme pairs, it could be determined that "M" and "N" are similar enough for "frame" and "rain" to be a good quality imperfect rhyme.

The work by Hirjee and Brown[8] offers a method to determine how well phonemes match. In this paper, ARPABet phoneme pairs are given a log-odds score that indicates how likely a pair is to co-occur. They are stored in two matrices; one for vowels and one for consonants. These scores for phoneme pairs, can be used to calculate a rhyme score for syllables, words, or sentences. The score for a pair of syllables is calculated as follows:

$$vowelScore + stressScore + consonantScore$$

Here, the vowelScore is the score found in the vowel matrix for the vowels. The stressScore is based on the stress indication of the vowels. When vowels have a matching strong stress, the score will be high. Contrary, when they have no strong stress or there is no match in stress, the score will be low. The consonantScore is based on the entries in the consonant matrix for the consonants that come after the vowel. Before the consonants are scored, they are aligned so that similar consonants are scored with one another. For example, when the words "code" and "mold" are scored, the consonants are aligned as "_ d" and "l d". This is done because identical consonants occurring in the end consonants of two words makes them more likely to rhyme and by matching them with each other this is also reflected in the scoring.

2.2 Rap lyrics generators

Other research is done that created tools that can generate rap lyrics. Some of the found tools are described in this section. The first example is DeepBeat[10] which is a tool that approaches the problem of lyrics generation as the selection of a new line out of a large data set. First, it extracts features (e.g. end rhyme, line length, etc.) out of the input line. Then, it ranks the possible lines on these similar features and picks the best one. According to the paper, the tool achieves a rhyme density 21% higher than most human rappers. Although this lyrics generator produces a high rhyme density, it also has two problems. First, this lyrics generator does not come up with original ideas because the lines it chooses already exist. This could still serve the purpose of creative stimulation but the generated lines can not be used due to plagiarism. Second, their way of determining rhyme is only based on vowels. So, if the vowels of words are the same, they are said to rhyme. This does not take into account that consonants make a significant distance for good quality imperfect rhyme. For example, break and shame would not be classified as rhyme by most people even though they share the same vowel sound.

Another example is GhostWriter[15], which tries to replicate the style of an artist on which the model was trained, whilst still creating new lyrics. It tries to create lyrics that imitate style by having a similar rhyme density, whilst minimizing the semantic similarity score of generated lyrics to the lyrics it was trained on. The model works in an unsupervised manner, meaning that it has no prior knowledge about phonetics or rhyme. Instead, it uses <endLine> and <endVerse> tokens in the training data to detect end rhyme. This enables the detection of rhyme schemes and generates rhyme based on that.

DeepBeat and GhostWriter have good capabilities in generating coherent rap lyrics however they do not stimulate creativity. DeepBeat uses existing lines and can therefore not come up with new ideas. GhostWriter does not stimulate creativity because it does not take user input and can therefore not respond directly to a lyric. There are tools that do stimulate creativity. For example, Freestyle[20], this tool creates a rhyming response to an input sentence. However, similar to GhostWriter, it does not use prior knowledge about phonetics but uses an unsupervised rhyme scheme detection[2]. This produces some perfect end rhyme, which is difficult for an unsupervised system. However, it does not produce intricate multisyllabic rhyme.

Rapformer[12] is another tool found in literature that produces rap lyrics from an input. However, instead of the input being lyrics it is another piece of media like a news article, for example. Therefore, this tool does not function as a good sparring partner as it just transforms the input instead of generating a follow up. An interesting aspect of this research is the usage of Bidirectional Encoder Representations from Transformers (BERT). BERT is a bidirectional language representation model meaning that it takes into account both the words before and after a given word. BERT can be used for masked language modelling. In a sentence, any word can be masked and BERT can fill this mask with a logical word. In Rapformer[12], words that need to rhyme are masked and then filled with rhyming words.

Separate to research done in computational creativity with regard to rap lyrics generation, it could also be done with AI models trained for general tasks. An example would be Chat-GPT[13], a conversational AI created by OpenAI, that can give answers to prompts from users. For example, a user can input it with "Generate a rap lyric of one line that continues on the following sentence:" followed by any desired rap lyric and it will produce a sentence with a thematic connection that also rhymes. However, just as the other examples this produces very little internal or multisyllabic rhymes. Nonetheless Chat-GPT is quite promising, it is also quite new and little research is done on how Chat-GPT can be used in computational creativity.

3 METHODOLOGY

In this section, the development of the lyrics generator is discussed along with the selection criteria of lyrics to be evaluated and how these lyrics were evaluated.

3.1 Developing the lyrics generator

The developed program[19] consists of several pieces, namely: Corpus, Bi-grams of text, Word Embeddings, Rhyme Detection, Word selection, and BERT. All of these components were visualised in figure 1 and will be explained in the following subsections.

3.1.1 Corpus. The program needs a corpus of lyrics in order to be able to generate lyrics. For this, a data set from RapLyrics-Scraper[14] was used. The scraper takes as input artist names and the amount of songs that are desired to be scraped. The used data set contains song lyrics from lyrics site "Genius" with at least 36 different rap artists.

3.1.2 Bi-grams of Text. The used dataset was split into n-grams, which are n subsequent words in a text. This was done because the program should be able to detect multisyllabic rhyme, which can span multiple words. For this research, bi-grams were chosen, which are n-grams with a size of two for n. For the bi-grams, the semantic meaning was also stored using word embeddings.

3.1.3 Word embedding. A word embedding is a representation of the meaning of a word. To achieve a word embedding, the meaning of a word is encoded into a vector of values between -1 and 1. Words that are close in vector space have a similar meaning. The word embeddings for individual words are gained from a FastText word vector set which contains 1 million words from wiki news [11]. For the Bi-grams, the mean of the two word embeddings is taken.

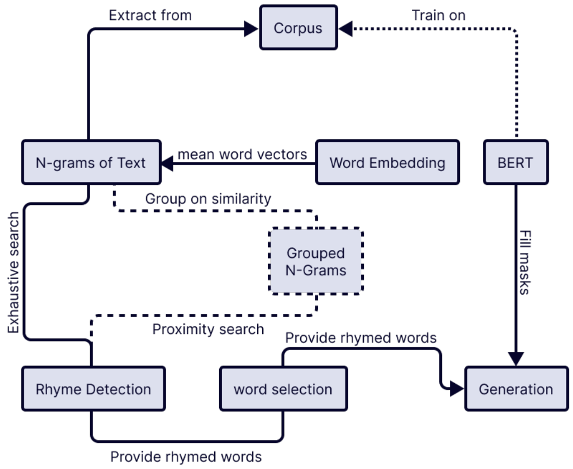

Fig. 1. Diagram showing how the components of the lyrics generator work together. The dotted lines represent components and relations that are not implemented in the current implementation.

3.1.4 Rhyme detection. The rhyme detection gives scores to syllables on how well they rhyme in a similar fashion to the work down by Hirjee and Brown[8]. What is different is that Hirjee and Brown group syllables by evenly distributing consonants over vowels. In the current study, syllables are split up using an adapted version of the CMU pronunciation dictionary [4]. This produces more accurate syllabification of words which improves the detection of multisyllabic rhyme.

In order for the program to be able to detect multisyllabic rhymes that span multiple words, a sliding-window was implemented. The words that need to be scored on their rhyme are first converted into their phonemes. Then, the size of the shortest group of syllables is taken as the "window size". Of the larger syllable group, the first syllables up to the window size are scored on their rhyme, this is stored and then the window is shifted by one. This repeats until the window reaches the end. Then, the highest rhyme score is returned. If there are more words that need to rhyme, the selection of rhymed words also uses the word embeddings of the bi-grams to select semantically similar rhyme words to ensure a thematic connection within the output sentence.

3.1.5 Word selection. The word selection function always selects the last word in a sentence. Furthermore, it selects extra words within the sentence up to the amount of syllables specified in the parameter of the function. It does not select a word if it is deemed not interesting (e.g "that", "I", "you"). This is done by checking whether the word is in a list of less interesting words. For these words, the

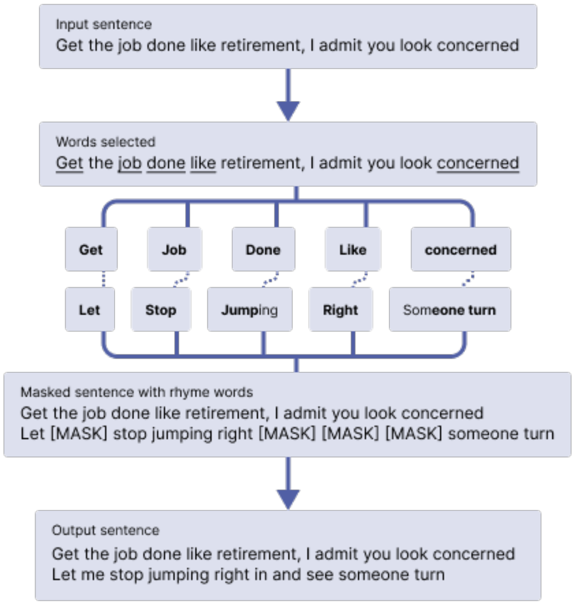
Input sentence
Get the job done like retirement, I admit you look concerned
Words selected
Get the job done like retirement, I admit you look concerned
Get
Job
Done
Like
concerned
Let
Stop
Jumping
Right
Someone turn
Masked sentence with rhyme words
Get the job done like retirement, I admit you look concerned
Let [MASK] stop jumping right [MASK] [MASK] [MASK] someone turn
Output sentence
Get the job done like retirement, I admit you look concerned
Let me stop jumping right in and see someone turn

Fig. 2. Follow through diagram of programs functionality when inputting the sentence "Get the job done like retirement, I admit you look concerned"[16]

rhyme detection component is used to look for rhyming n-grams and feed those back to the generation component.

3.1.6 BERT. As mentioned, BERT does masked language modelling, meaning that it can fill mask tokens in a sentence with logical words. The BERT component will get rhyme words from the generation component padded with mask tokens. It will fill these masks to produce the output.

3.1.7 Generation. The generation component takes an input lyric and uses the previously mentioned components to generate lyrics. It starts off by selecting interesting words to rhyme. Then, it selects from the bi-gram database a set of words that rhyme and have a semantic connection with each other. Those words are put in a sentence that padded with BERT mask tokens in between them. The padding between rhymed words is based on the amount of words between selected words to rhyme from the input lyric. This then gets fed to BERT, which will try to fill in the mask tokens to generate a logical sentence. In diagram 2, a follow through example is given of this process.

3.2 Selecting lyrics

The lyrics generator was evaluated by presenting people with input line and a generated output line of the application as well as existing sets of two lines of lyrics. The selection criteria for the existing lyrics are as follows:

- The selected lyrics should contain multiple syllables that rhyme. This should not only include end rhyme.
- The lyrics should not be from a popular song and should not be from the hook or chorus of a song. This way, the participant taking the survey is less likely to recognize the lyrics, and distinguish them from computer generated ones.
- The lyrics should not contain any profane words or discriminatory slurs.

The generated lyrics should follow the following selection criteria:

- The lyrics should not contain any profane words or discriminatory slurs.
- The first sentence comes from the same songs and artists chosen for the existing lyrics. The second sentence is generated by feeding the first sentence as input to the program.
- There should be no major human interference with the generated output. However, minor changes like capitalization, or punctuation marks can be adapted.

The lyrics that were evaluated in the survey can be found in appendix A. The artist denoted in the "Artist" column is not necessarily the main artist credited on the song but instead the artist rapping the specific verse of that song. If the column "Generated" denotes "Yes", the first line is picked from that artist and the second line is generated by the tool.

3.3 Evaluating the lyrics generator

The survey includes demographic, quantitative, and qualitative questions. In the demographic questions, the gender, age range, English proficiency, and acquaintance with rap music of the participant were asked.

In the quantitative analysis, the participant was presented with ten stanzas of rap lyrics. Half of the stanzas were existing song lyrics. The other half consisted of an existing song lyric from the same songs paired with a generated follow-up lyric. Participants scored all stanzas on two aspects, namely rhyme quality and coherence. Here, coherence indicates that the output sentence logically follows from the input sentence by having a relating theme, sentiment or style to the input sentence. The scoring was done through a Likert-like seven point scale. For the rhyme quality, a score of one is labelled with "poor rhyme" a score of seven is labelled as "good quality rhyme". For the coherence, a score of one means that "the lines have nothing in common" whereas a score of seven means "the lines are coherent and continue on in the same theme, style or sentiment".

In the qualitative analysis, the participant are presented with the same five generated stanzas. The rhymes that the program has chosen will be highlighted as well. Here, the participants were asked to write a short paragraph on both the rhyme quality and coherence.

4 RESULTS

In the survey, 40 participants took part of which one has been excluded because they did not tick all of the checkboxes for informed consent. Four participants were excluded from the quantitative analysis because they did not answer all questions, however, they were included in the qualitative analysis. The participants were a

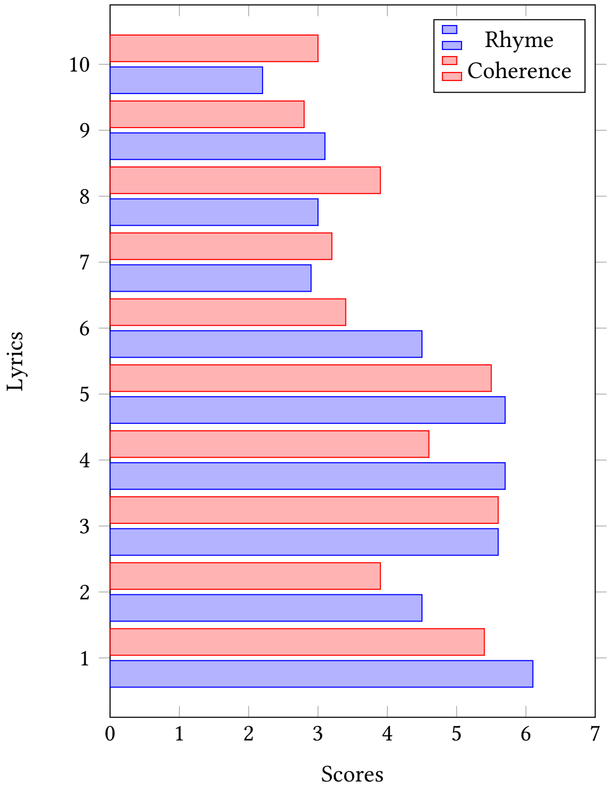

Fig. 3. Average rhyme and coherence scores for lyrics

convenience sample of acquaintances of the researcher. Out of the participants, 28 identified as male, 9 as female and 2 as non binary. Out of the participants, 18 were in age-range 18-21, 17 in 22-25, 3 in 26-29 and 1 in 30+. Furthermore, a good understanding of the English language was captured in the survey. On average, participants rated themselves a 4 out of 7 on how much they voluntarily listen to rap music. Lastly, 28 participants had never written rap lyrics, 3 people have written lyrics before but not rap lyrics, and 8 have written rap lyrics.

4.1 Quantitative analysis

In chart 3 the average scores of lyrics are displayed. The lyrics numbered 1 through 5 are the existing texts whilst the lyrics numbered 6 through 10 are the generated texts. The existing lyrics score, on average, 5.0 in coherence and 5.52 in rhyme score. The generated lyrics score, on average, 3.26 in coherence and 3.14 in rhyme score. Meaning that the existing lyrics average score was higher than the generated lyrics on both rhyme quality and coherence.

Sums of all scores given by a participant were made and grouped in the following four groups: rhyme existing, rhyme generated, coherence existing, and coherence generated. These groups were tested on whether they were normally distributed according to the Shapiro-Wilk test. The Shapiro-Wilk test showed that all groups were normally distributed.

To test the statistical difference between the scores of generated and existing lyrics, A paired t-Test has been conducted. The existing rap lyrics score significantly higher than the generated rap lyrics on rhyme, t(34) = 13.83, $p < .001$ and on coherence, t(34) = 11.40, $p < .001$. This means the null-hypothesises, that there are no differences in rhyme and coherence between existing lyrics and generated lyrics, are rejected.

4.2 Qualitative analysis

From the qualitative assessment it becomes evident that some opinions vary widely. In the following sections, these will be discussed per lyric.

4.2.1 Lyrics #6. The part "go to figure" is seen as illogical by a large portion of the participants. The coherence is present, mainly due to the use of colors ("gold" and "dark"). The input sentence is appreciated by most participants, which makes some of them disappointed in the output sentence.

4.2.2 Lyrics #7. There is some coherence because of the words "world" and "vision". Another participant says that the lines work together even though they do not really rhyme. The sentence contains the word "gonna" twice after one-another. Participants found this to be bad and distracting. The words "shame" and "change" are not seen as nice rhyme words.

4.2.3 Lyrics #8. The opinion on "nosebleed" and "nose be" vary. Some participants appreciate it, others dislike it. Some also think it does not rhyme since they both contain the word "nose". The opinions about the rhyme words "going" and "in" vary too. Participants also see the second line as gross or confusing. One participant thinks the generated output is funny. In general, the rhymes are described by the participants as surprising.

4.2.4 Lyrics #9. Using the word "T's" causes some confusion among participants. Some people see it as creative or clever. Others think it does not make sense. The words "now" and "out" are also not seen as rhyming by everyone. A participant thinks the lyric might work in the context of a larger song.

4.2.5 Lyrics #10. The words "All" and "Y'all" are seen as nice by most people, others think it is lazy. The grammar does not make much sense, but a few participants think that with an extra word like "can" or "go" it would actually be one of the best lyrics generated. Theme-wise the word "insanity" is paired to the 9-5 mentality hinted at in the output sentence.

4.2.6 General. Some rhymes are not seen as rhyme. This makes sense for half rhymes, however perfect rhymes like "keep" and "sleep" and "T's" and "peace" are also not seen as such. The generated outputs are often criticized on their coherence due to the poor grammar. The end-rhyme could be stronger on most instances. Some participants found it challenging that they only had two lines to look at. They felt like a bigger picture was needed to understand whether lines are coherent or not. They did not only feel about this for the generated lyrics but also the existing lyrics.

Contrary to the criticism, some participants are impressed that

AI is capable of generating rhymes and lyrics such as these. The tone of the generated lyrics are seen as nice. One participant said that the lines were "pretty funny and good in terms of sound". In general, participants see potential.

5 DISCUSSION

5.1 Explanation of results

The participants in the survey, are significantly more positive about the existing lyrics than the generated lyrics. This is to be expected since experienced rappers create good rhymes and coherence as a profession. Furthermore, the feedback of the participants largely aligns with the known limitations of the current implementation of the program. Even with the lower score on generated texts, participants recognize that the program has potential. This suggests that the proposed program can be used when it is improved.

6 LIMITATIONS AND FUTURE WORK

Firstly, the limitations of the current implementation are discussed. Secondly, in the Future Work section, it is discussed how these limitations could be improved in the current architecture.

6.1 Limitations

6.1.1 Scalability and speed. The current program does not scale well for looking up rhyme words. As of right now, a word is compared to every bi-gram in the data set. It is then sorted on rhyme score. This can take very long if the input grows larger. Ideally, a larger input is desired as it enables a higher rhyming vocabulary

6.1.2 BERT. The current implementation of BERT within this project is not trained further than the base model. Sometimes it can still generate correct sentences but occasionally it produces artifacts such as repeating of words and filling masks with punctuation marks instead of logical words. Furthermore, transformers are known to be good at replicating style if they are trained on the right data [12]. Currently, BERT does not generate text in the style of rap lyrics.

6.1.3 Key Errors. The current implementation of the program uses dictionaries for word embeddings and for phoneme transcription of words. Rap lyrics can have unique words, abbreviations or names in there that are not in the keys of the dictionaries.

6.1.4 Rhyme and Rhythm. In general, the rhyme detection works as it should. However, two properties of multisyllabic rhyme are not fully realised within the code. Firstly, in multisyllabic rhymes a syllable in the middle of the rhyme can be skipped (eg. "last feather" and "fast like weather"). This does not work with the current sliding window approach. Secondly, as mentioned in the introduction, multisyllabic rhyme can span multiple words. In the current implementation, one word of the input can rhyme with two words from the bi-grams but the inverse relation can not be found. Also, rhymes with more than two words can not be detected.

Although the CMU dictionary provides a way to check rhythm with stressed and unstressed syllables, this was not done in the current implementation of the program, largely due to the masking done with BERT. BERT is used to strive towards grammatical correctness in the output sentence. But often times, the words that are best for the grammatical structure of the sentence might contain too little or too many syllables. Because it can not be controlled how many syllables BERT produces, there is no control over the amount of syllables that are put in the generated output sentence.

6.2 Future Work

6.2.1 Local sensitivity hashing. The scope of this research was to develop and evaluate a lyrics generator. However, there are things that can be done to improve the efficiency. As mentioned, selecting the best rhyming bi-gram is done through comparisons with all bi-grams. This means that for one word that needs to rhyme, the entire selection of bi-grams has to be traversed. One way to speed this up is through local sensitivity hashing. This means grouping similar words together, so that words only have to be compared with words that are within their group. If this were to be incorporated, the amount of bi-grams could grow without impacting the run time of the program in a significant way, thus increasing the vocabulary of the AI. It would also allow for the bi-grams to be replaced with tri-grams to enable rhymes across three words. Furthermore, it would also make sense to use local sensitivity hashing as it groups words that are similar, and similarity in pronunciation is what categorizes words that rhyme.

6.2.2 Training. BERT can be trained with a corpus of rap lyrics. For example, the corpus used for the bi-grams or an extended corpus that can be made through RapLyrics-Scraper[14]. With a better trained BERT model, the generator will produce an output that is more like rap lyrics. Further training of BERT might also limit the amount of artifacts in the generation.

6.2.3 Replacing dictionaries with models. The CMU dictionary and word embedding dictionary can be replaced with models that classify words. An example for word embeddings would be fast text that can create word embedding based on sub-words[5]. Whilst the current implementation also takes the word embeddings from fast text it does not implement the sub-word methodology. The CMU-dictionary can be replaced by a grapheme-to-phoneme model (g2p). This is a model trained on generating a phonetic transcription from a grapheme (spelling) of a word.

6.2.4 Better rhyme detection. In order to detect a gap in multisyllabic rhyme, some sort of alignment should be performed which increases the computational cost. This alignment can be done based on rhyming vowels the way that it is also done in the work by Hirjee and Brown[8]. Additionally, the current text generation does not incorporate alliteration, which is rhyme that occurs before the vowel instead of after. An example of this would be the tongue twisters like "Sally sells seashells by the sea shore". The matrix by Hirjee and Brown[8] only has scores for the consonant types that occur after a vowel. So, in order to account for alliteration, a new matrix would have to be calculated.

7 CONCLUSION

In this paper, an architecture for a rap lyrics generator focused on multisyllabic rhyme has been proposed. A working prototype of this architecture has been created. However, there are several limitations to the implementation of the program. The prototype generated five stanzas that were compared with existing lyrics. These stanzas were put in a survey and evaluated both on a quantitative level and a qualitative level. On the quantitative level, the stanzas were scored on rhyme and coherence. The generated lyrics scored lower than the existing lyrics. In the qualitative part, some of the limitations of the generated output were mentioned. These largely coincide with the known limitations of the current implementation. Furthermore, the participants addressed that with some work or in some context the lyrics could work. This leads to the conclusion, that with some extra work to the program, a high quality multisyllabic rap lyric generator could be established using the proposed architecture in this research with improved components. This lyric generator can in turn be used to stimulate the creativity of musicians.

REFERENCES

[1] Kendrick Lamar A Tribe Called Quest. 2016. Conrad Tokyo.

[2] Karteek Addanki and Dekai Wu. 2013. Unsupervised rhyme scheme identification in hip hop lyrics using hidden Markov models. In *Statistical Language and Speech Processing: First International Conference, SLSP 2013, Tarragona, Spain, July 29-31, 2013. Proceedings 1*. Springer, 39–50.

[3] J. Cole Ari Lennox. 2019. Shea Butter Baby.

[4] Susan Bartlett, Grzegorz Kondrak, and Colin Cherry. 2009. On the syllabification of phonemes. , 308–316 pages.

[5] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching Word Vectors with Subword Information. https://doi.org/10.48550/ARXIV.1607.04606

[6] Danny Brown. 2016. Ain't it Funny.

[7] Paul Edwards. 2009. *How to rap*. Chicago Review Press. 81–91 pages.

[8] Hussein Hirjee and Daniel G Brown. 2009. Automatic Detection of Internal and Imperfect Rhymes in Rap Lyrics.. In *ISMIR*. International Society For Music Information Retrieval, Kobe, Japan, 711–716.

[9] Kevin Lenzo et al. 2007. The CMU pronouncing dictionary. *Carnegie Melon University* 313 (2007), 74.

[10] Eric Malmi, Pyry Takala, Hannu Toivonen, Tapani Raiko, and Aristides Gionis. 2016. DopeLearning: A Computational Approach to Rap Lyrics Generation. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (San Francisco, California, USA) *(KDD '16)*. Association for Computing Machinery, New York, NY, USA, 195–204. https://doi.org/10.1145/2939672.2939679

[11] Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in Pre-Training Distributed Word Representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

[12] Nikola I. Nikolov, Eric Malmi, Curtis Northcutt, and Loreto Parisi. 2020. Rapformer: Conditional Rap Lyrics Generation with Denoising Autoencoders. In *Proceedings of the 13th International Conference on Natural Language Generation*. Association for Computational Linguistics, Dublin, Ireland, 360–373. https://aclanthology.org/2020.inlg-1.42

[13] OpenAI. 2022. Chat-GPT. https://chat.apps.openai.com/

[14] François Paupier. 2021. RapLyrics-Scraper. https://github.com/fpaupier/RapLyrics-Scraper.

[15] Peter Potash, Alexey Romanov, and Anna Rumshisky. 2015. GhostWriter: Using an LSTM for Automatic Rap Lyric Generation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, 1919–1924. https://doi.org/10.18653/v1/D15-1221

[16] the Creator Tyler. 2019. NEW MAGIC WAND.

[17] Mad Villain. 2004. Accordion.

[18] Daniel A. Walzer. 2017. Independent music production: how individuality, technology and creative entrepreneurship influence contemporary music industry practices. *Creative Industries Journal* 10, 1 (2017), 21–39. https://doi.org/10.1080/17510694.2016.1247626 arXiv:https://doi.org/10.1080/17510694.2016.1247626

[19] Matthias Wentink. 2023. Rap Lyrics Generator. https://github.com/MatthiasWentink1/Rap-Lyrics-Generator.

[20] Dekai Wu, Karteek Addanki, Markus Saers, and Meriem Beloucif. 2013. Learning to Freestyle: Hip Hop Challenge-Response Induction via Transduction Rule Segmentation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Seattle, Washington, USA, 102–112. https://aclanthology.org/D13-1011

Appendices

A LYRICS USED IN THE SURVEY

#	Artist	Generated	Lyric
1	Kendrick Lamar[1]	No	Toleration for devastation, got a hunger for sin Every nation abomination, let the coroner in
2	J. Cole[3]	No	You crossed my mind a thousand times The cost was fine, I draw the line
3	Tyler, the Creator[16]	No	Take one look in the mirror, implications so clear. I live life with no fear, except for the idea that one day you won't be here
4	Danny Brown[6]	No	Verbal couture, parkour with the metaphors The flow house of horror, dead bolted with metal doors
5	MF Doom[17]	No	Living off borrowed time the clock ticks faster That will be the hour they knock the slick blaster
6	MF Doom[17]	Yes	Keep your glory, gold and glitter Sleep dark and go to figure
7	Tyler, the Creator[16]	Yes	And let the world know 'cause I ain't got no shame I get to remember whole vision's gonna gonna change"
8	Danny Brown[6]	Yes	Nosebleed on red carpets, but it just blend in Nose be gone get the thing going
9	J. Cole[3]	Yes	I pray you found peace and you whole now The scales usually out T's and come out
10	Kendrick Lamar[1]	Yes	And manifest all insanity, look around Y'all can to work now