

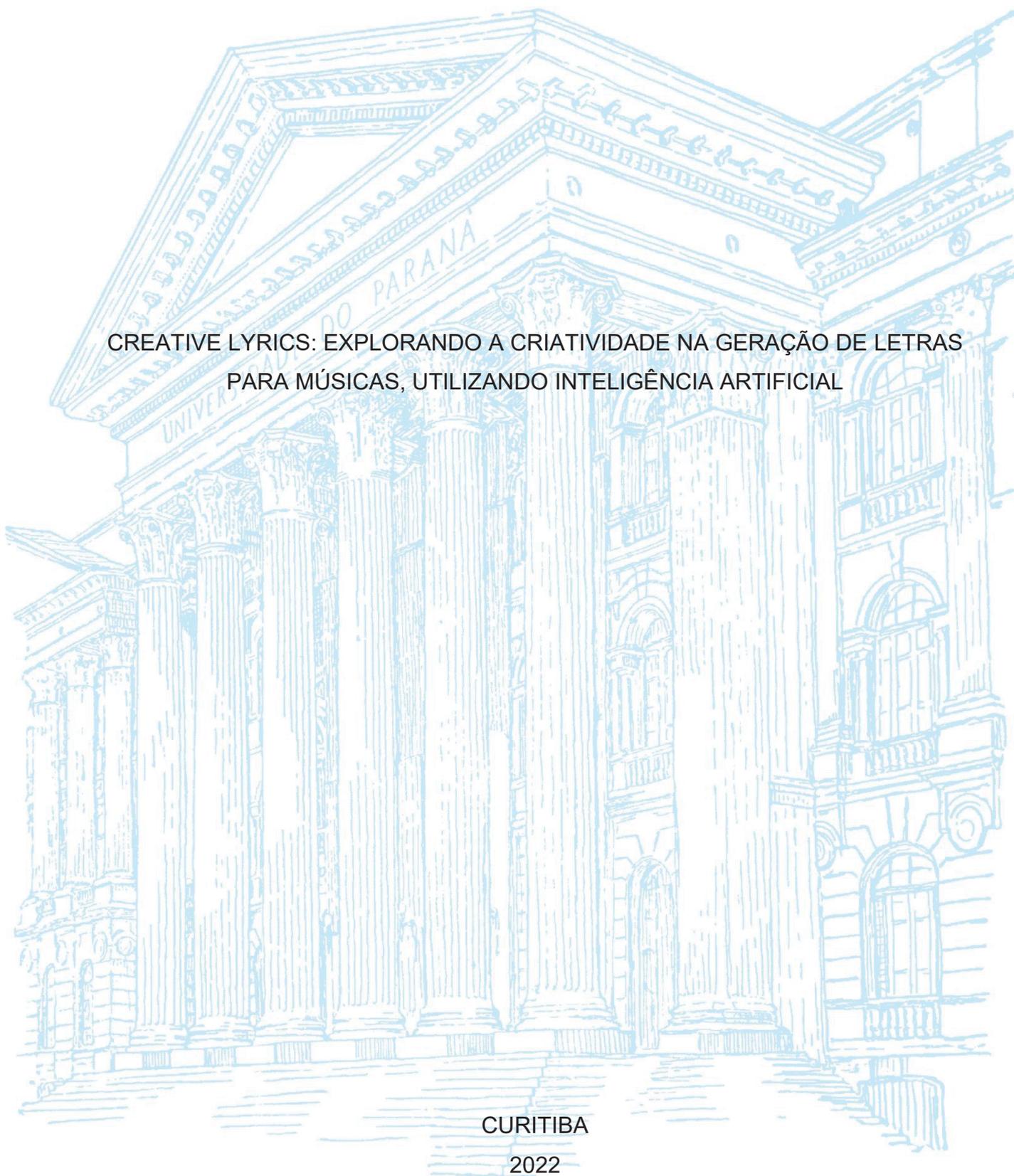
UNIVERSIDADE FEDERAL DO PARANÁ

JOSÉ AUGUSTO DA SILVA FREITAS

CREATIVE LYRICS: EXPLORANDO A CRIATIVIDADE NA GERAÇÃO DE LETRAS
PARA MÚSICAS, UTILIZANDO INTELIGÊNCIA ARTIFICIAL

CURITIBA

2022



JOSÉ AUGUSTO DA SILVA FREITAS

CREATIVE LYRICS: EXPLORANDO A CRIATIVIDADE NA GERAÇÃO DE LETRAS
PARA MÚSICAS, UTILIZANDO INTELIGÊNCIA ARTIFICIAL

Trabalho de Conclusão de Curso apresentado ao curso de Pós-Graduação em Inteligência Artificial Aplicada, Setor de Educação Profissional e Tecnológica, Universidade Federal do Paraná, como requisito parcial à obtenção do título de Especialista em Inteligência Artificial Aplicada.

Orientador: Prof. Dr. Lucas Ferrari de Oliveira

CURITIBA

2022

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação INTELIGÊNCIA ARTIFICIAL APLICADA da Universidade Federal do Paraná foram convocados para realizar a arguição da Monografia de Especialização de **JOSE AUGUSTO DA SILVA FREITAS** intitulada: **Creative Lyrics: Explorando A Criatividade Na Geração De Letras Para Músicas, Utilizando Inteligência Artificial**, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de especialista está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 10 de Novembro de 2022.



LUCAS FERRARIDE OLIVEIRA
Presidente da Banca Examinadora



RAZER ANTHOM NIZER ROJAS MONTANO
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

Creative Lyrics: Explorando a criatividade na geração de letras para músicas, utilizando Inteligência Artificial

José Augusto da Silva Freitas
Setor de Educação Profissional e Tecnológica
Universidade Federal do Paraná
Curitiba, Brasil
guto.sonora@gmail.com

Lucas Ferrari de Oliveira
Departamento de Informática (DInf)
Universidade Federal do Paraná
Curitiba, Brasil
lferrari@inf.ufpr.br

Resumo—A música e o empreendedorismo, estão conectados diretamente pelo processo criativo. A economia criativa tem sido uma importante atividade econômica e com o uso da tecnologia tem dado velocidade e continuidade na produção de conteúdo. O processo de composição é alimentado por criatividade, mas realizar as composições de música envolvendo letra, melodia e uso de instrumentos, pode se tornar muitas vezes complexo. O estudo apresentado, demonstra como a IA pode ajudar na geração de palavras, simulando uma composição de letras de música de um determinado estilo musical ou tema, neste caso *Heavy Metal* e *Melodic Death Metal*. O objetivo é auxiliar e inspirar o compositor no processo criativo. A pesquisa foi realizada utilizando os processadores de texto do *framework TensorFlow*, com uma base de dados de letras criadas a partir de um processo de *web scraping*. Foram utilizados os modelos de Redes Neurais Recorrentes, *Simple RNN* combinadas como o modelo de *Markov*, com a linguagem *Python*. Foi possível aplicar algumas técnicas de tratamento da informação com o pré-processamento, o que após aplicação do modelo trouxe bons resultados de geração de letras. Com ou sem um tema direcionado, é possível utilizar IA também para auxiliar nesta importante etapa de composição.

Palavras-chave—música, tecnologia, letras, criatividade, geradores de texto, IA, tensorflow, heavy metal, melodic death metal

Abstract—Music and entrepreneurship are directly connected by the creative process. The creative economy has been an important economic activity and the use of technology with continuity in the production of content. The songwriting process is fueled by consistency, but performing complex music, melody, and instrument compositions is something often. The study presented, demonstrated how AI can help in the generation of words and lyrics, simulating a composition of song lyrics of a certain musical style or theme, in this case *Heavy Metal* and *Melodic Death Metal*. The goal is to help and inspire the composer in the creative process. The search was used to use processing text from the TensorFlow framework, with a database created by webscraping process merging lyrics. The models of Recurrent Neural Networks, Simple RNN, combined with the Markov model, with the Python language were used. It was possible to carry out several steps of processing the information with the pre-processing, which after the application of the model, brought good results in the generation of letters. With or without a targeted theme, you can also use AI to help with this important composing step.

Index Terms—music, technology, lyrics, creativity, text generators, AI, tensorflow, heavy metal, melodic death metal.

I. DESENVOLVIMENTO

Embora a música esteja relacionada com arte e indiretamente ao processo criativo, historicamente ser musicista, compositor ou intérprete, só passou a ser reconhecido como atividade profissional após a Segunda Guerra Mundial [1]. A música, e o empreendedorismo, possuem uma relação muito importante, pois ambos os conceitos dependem de um processo criativo. E a economia criativa, tem sido uma relevante atividade econômica com grande potencial de oferta para serviços, produtos e também por envolver mais de 5,2 milhões de pessoas em atividade apenas no Brasil [1] [2].

Paralelo a este cenário, a tecnologia tem influenciado diretamente o universo musical e o empreendedorismo. A medida que a influência da tecnologia avança sobre o contexto musical, é importante considerar que existe uma necessidade de velocidade e continuidade na produção de conteúdo, e que representem em forma de canções a cultura, a época, os sentimentos e as atitudes do ser humano [2] [3].

Realizar toda a composição de uma música, envolvendo a letra, a melodia, e uso de instrumentos, transformando em algo sonoro, é algo complexo [4]. Porém é possível utilizar reconhecimento de padrões e repetições que podem ajudar na construção da sua estrutura, tanto na parte instrumental, quanto na parte lírica utilizando técnicas musicais, modelos matemáticos e IA para facilitar o desenvolvimento de novas canções [4] [5].

Neste trabalho, iremos explorar como a tecnologia por meio da IA, pode contribuir com o processo criativo, explorando a criação de letras de músicas a partir de processadores e geradores de texto. O objetivo principal, é auxiliar o compositor por meio de expressões que possam inspirar ou até mesmo facilitar o processo de criação da letra, de acordo com tema ou artista escolhido.

Embora a música seja influenciada pela cultura, tempo e outros aspectos artísticos, auxiliar mesmo que em uma parte da composição, no caso a letra, pode ajudar o compositor ter continuidade no processo criativo [1] [2].

A. Descrição dos dados

Para coletar os dados para análise e construção do modelo, foi necessário um processo de *Web Scraping* conhecido como raspagem, programado em *Python* [6]. A etapa inicial foi procurar um *site* publicado com estrutura e conteúdo adequado para esta captura, neste caso o *AZLyrics*¹.

O processo consistiu em identificar os artistas relacionados a um estilo e tema de interesse do autor e com isso listar todas as letras disponíveis no *site*. Depois, pesquisando link a link e copiando o retorno do endereço *html* para uma base de dados. Por meio desta ação, permitiu-se uma análise prévia da estrutura das letras, a exploração de características nelas presentes, ajustes dos campos, etapas de limpeza e pré-processamento. As etapas foram realizadas a partir da exportação para o software *Excel* e novamente para o *Python* [6].

Ao buscar pelos dados, verificou-se que as letras podem variar de maneira ampla de acordo com o estilo e linguagem, bem como por outras características sócio-culturais [1] [2]. Para o estudo, foi importante definir uma linguagem de idioma específico, sendo escolhido então a língua inglesa. Também houve a escolha de um grupo de estilo determinado, neste caso sendo bandas de *Heavy Metal* e *Melodic Death Metal*, por serem estilos semelhantes [7]. Curiosamente 30% das bandas relacionadas são da Finlândia. De acordo com Urtaza [7], isso pode ser explicado devido ser o país como maior número de bandas deste estilo per capita do mundo.

Após o processo de coleta destas informações, e limpeza dos dados, foi necessário classificar e agrupar as letras. Isso para compreender melhor a estrutura de cada canção e escalar em um nível mais amplo para compreender suas principais características, sua fluência e principalmente adequar estas informações para uso do gerador de texto. Interpretando estes dados iniciais foi possível mapear algumas informações relevantes para estudo pós pré-processamento:

- 3.250 Letras de Músicas;
- 28 Artistas/Bandas;
- 531.275 total de palavras encontradas;
- 20.384 palavras únicas;
- 164 palavras em média por música;
- 3.331.560 caracteres;
- 120 caracteres únicos.

Com esta etapa inicial, foi possível criar um *dataframe* contendo o identificador da música, o nome da música, link de extração no formato *html*, artista/banda e letra da música, como mostra a Figura 1.

A principal dificuldade encontrada, além de melhorar o sentido e compreensão da mensagem final gerada, foi entender e aproximar a fonética das palavras para interpretação e geração das rimas. Ou seja, de acordo com o final das sentenças, é necessário (mas não obrigatório) que exista uma rima para facilitar a fluência da música [1] [7]. Para rima em específico, percebeu-se que o tempo para compreensão e entendimento

ID	SONG_NAME	LINK_URL	ARTIST	LYRIC
1	Leaden Legacy	lyrics/afterforever/leadenlegacy.html	After Forever	I can't look behind because I'm living in the past Tears without emotion are filling up the unbridled hours We cannot be What you expect from us We cannot see through the glasses of despair Evade the leaden legacy Those who separate and join the opposite do not see that that will get them nowhere Can't you understand it is a similar manipulation Wake up and see the light Doomed to wander in repressed emotions This imaginary refuge restrains me from awakening We cannot be What you expect from us We cannot see through the glasses of despair Evade the leaden legacy Those who separate and join the opposite do not see that that will get them nowhere Can't you understand it is a similar manipulation
2	Semblance Of Confusion	lyrics/afterforever/semlanceofconfusion.html	After Forever	As the light's becoming more fierce And the heat is mel
3	Black Tomb	lyrics/afterforever/blacktomb.html	After Forever	The storm stings through the mountains While the rain is

Figura 1. Exemplo dos primeiros registros coletados e armazenados via processo de *web scraping*.

da fonética ou opções de representação para troca por substantivos, seriam melhor avaliados em um estudo dedicado no futuro. Outro ponto importante é qual técnica poderia trabalhar mais facilmente com tema e quando exatamente trabalhar esta técnica? Com base em alguns trabalhos já publicados envolvendo processamento de texto e processamento de Linguagem Natural conhecida como *NLP*, a indicação foi iniciar com Redes Neurais Recorrentes [1] [8] [9].

Embora o objetivo seja auxiliar o compositor no processo criativo, a entrega de palavras próximas, mesmo quem em alguns momentos não façam sentido, podem facilitar na inspiração do processo criativo de composição.

B. Métodos

Para exploração deste tema de pesquisa, foram desenvolvidos 4 macro etapas para execução. Conforme representado pela a Figura 2, cada uma delas foi responsável por uma parte importante no processo. A primeira já citada anteriormente, correspondeu a fase de captura da informação e entrega da base de dados para ser processada. A segunda etapa consistiu no processamento de dados, responsável pela definição de abordagem e também pela *tokenização*, transformação e vetorização. A terceira etapa, foi responsável pela criação do modelo a ser treinado. Ela também ajudou a determinar quais seriam as bibliotecas e *frameworks* a serem utilizados para predição, como descrito a seguir. Por fim a última etapa correspondeu a entrega do gerador de texto, com a execução dos modelos criados para comparação e análise textual da letra de música gerada.

Vale ressaltar que os modelos de linguagens são muito bons para interpretar a semântica e sintaxe com precisão e com isso obter o contexto das palavras, mas mesmo com esta precisão, os resultados podem ser insatisfatórios por serem genéricos, monótonos ou até mesmo repetitivos [10].

1) *Pré-Processamento dos Dados*: A fim de garantir uma maior qualidade nos dados, e para ajudar no processo de interpretação das informações, é necessário realizar um pré-processamento [11]. Isso permitirá que o modelo a ser trabalhado processe a informação (*input*) de maneira eficiente sem prejudicar o aprendizado de máquina ou análise do modelo praticado [11] [12].

¹Site AZLyrics Disponível em: <https://www.azlyrics.com>



Figura 2. Etapas no método aplicado no desenvolvimento do estudo.

O processamento de dados, foi inicialmente realizado com o objetivo de checar e entender a qualidade dos dados [13]. Após isso avançou com algumas ações e necessidades envolvendo processo de conhecimento dos dados e *Dataming* realizando a compreensão do idioma de todas as letras, entendimento dos cenários envolvendo o tipo de campo, caracteres especiais, abreviações, tratativa de nulos e principalmente na informação contida na letra do artista. Esta ação facilita entender os elementos e padrões mesmo que discretos, antes das etapas que envolvem vetorização ou mesmo aprendizado de máquina [12] [11].

Baseados nos critérios apresentados, foram eliminadas previamente 110 músicas que ou estavam com dados incompletos, com idiomas incompatíveis com o propósito (finlandês, alemão e outros dialetos), nulos ou que poderiam atrapalhar o interpretador. Após isso, foram mapeados 9 casos de letras em duplicidade também retiradas da base, que resultou em um total de 3.250 letras de músicas distintas e com o mesmo idioma.

Também foi executado uma transformação e retirada de delimitadores de texto, que pudessem alterar a compreensão das palavras.

Para interpretar os dados (depois da análise já realizada previamente com o software *Excel*), foi utilizado uma ferramenta conhecida para tratar e processar textos, a biblioteca

*NLTK*² do *Python*, realizando algumas operações importantes atreladas a outro *framework* conhecido para *machine learning*, o *TensorFlow*³, executando as principais etapas descritas a seguir:

- **Tokenização:** Consiste na divisão do texto em frações menores, como separar por palavras, grupos de palavras ou reduções que podem chegar até o nível de cada letra. A ideia foi utilizar este método (*word_tokenize*⁴) para possibilitar a quebra de cada frase com seu conjunto de *tokens* específico e já eliminando a pontuação e outros separadores indesejados.
- **Stemming:** Foi considerado também o uso *PorterStemmer*⁵, também presente na *NLTK*. Ele ajudou a interpretar palavras e sinônimos e substituiu grupos de palavras por semelhança e conjugação de tempo simples, pois reduz a palavra até a sua raiz gramatical removendo o prefixo ou sufixo. Isso foi realizado para posteriormente testarmos se a opção de geração de texto por palavras, seria mais interessante em relação a custo computacional ou qualidade na predição, do que pela subdivisão em caracteres. Uma opção considerada também, para aproximar as palavras em um mesmo contexto semântico foi o *Embedding*, que pode mapear as palavras pelo seu contexto, mas isso em uso conjunto ao processo de *Stemming* pode provocar redução no vocabulário final, pois todo contexto gramatical para geração de letras neste estudo é importante.
- **Limpeza e Transformação:** Foi utilizada outra biblioteca chamada *re*⁶, para trabalhar com expressões regulares e aplicação de *regex token* em cada frase, uniformizou as letras em minúsculas e removeu símbolos indesejados. Isso se faz necessário para melhorar a compreensão do vocabulário a ser criado. Ainda nesta etapa, foi utilizado um processo de reconhecimento da linguagem com a biblioteca *NLTK* para mapear o idioma das letras das músicas, afim de filtrarmos o banco de dados obtido e utilizar apenas as de língua inglesa;
- **Segmentação por Artista:** Devido a necessidade de entender melhor o contexto das músicas, visto que são artistas e temas diversos, foram criadas duas novas bases a partir da inicial. A primeira focando especificamente em uma banda com todas as suas composições onde dentre os 28 artistas pesquisados, foi definido para o estudo o grupo *Dark Tranquility*. A escolha baseou-se no volume de letras (168) e volume de palavras (28.075). Esperava-se que a variedade de letras e palavras, contribuísse para um melhor vocabulário;
- **Segmentação por Tema:** Em relação ao tema, buscou-se

²Biblioteca NLTK Disponível em: <https://www.nltk.org/index.html>

³*TensorFlow*. Disponível em: <https://www.tensorflow.org/>

⁴*Tokenize on TensorFlow*. Disponível em: https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/text/Tokenizer

⁵Documentação *PorterStemmer*. Disponível em: https://www.nltk.org/_modules/nltk/stem/porter.html#PorterStemmer.stem

⁶Expressões Regulares no Python. Disponível em: <https://docs.python.org/pt-br/3/library/re.html>

Tabela I
TEMAS DAS LETRAS

Temas	Quantidade
Government	127
War	437
Religion	590
Death	738
Love	794
Emotions	1.607

Tabela II
TRAIN SHAPE

	Train X	Train Y
Lyrics_Full	(137.805, 29)	(137.805, 20.384)

agrupar todas as 3.250 letras em 6 temas variados, separados conforme a ocorrência de determinadas palavras que representem este tema [14]. Embora não tenha sido localizada uma convenção definida para a classificação destes temas, foi criada uma lista de grupos derivados das palavras com maior ocorrência conforme mostra a tabela I, com o tema escolhido *Emotions* presente em 1.607 canções.

Embora algumas literaturas indiquem a remoção das *Stopwords*, por não representarem o contexto ou conteúdo da informação, para o contexto gramatical que se espera na geração das letras de música, todo conteúdo para sentença é importante [15]. Outro ponto de atenção foi em relação a vetorização do texto, neste caso *Word2Vec* foi também utilizado, mesmo com a aplicação anterior (*word_tokenize*).

2) *Base de Treino*: Optou-se pelo uso do modelo de *Markov*⁷ para gerar as palavras iniciais do gerador considerando todas as letras. Ele garante que o primeiro conjunto de palavras para cada frase tenha alguma coerência antes de re-alimentar a rede neural [16].

A ideia é filtrar o artista/banda ou o tema após o modelo já treinado. Mas observou-se que o isso seria ineficiente pelo fato de filtro ocorrer pós treinamento. Então foram realizadas fases de treino, considerando todas as músicas e as letras (*Lyrics_Full*), apenas de um artista definido (*Lyrics_Artist*) e do tema escolhido (*Lyrics_Theme*).

Com o uso do *TensorFlow*, foram criadas as bases de treino com *shape* demonstrado na tabela II, tal qual sua finalidade de uso esperada, seja total, artista ou tema, ou considerando a palavra inteira ou considerando a redução até o nível do caractere/letra. Foi aplicado antes de cada criação de base o uso do modelo *Markov* e após aplicação deste, também foi utilizado o exemplo de gerador de texto disponível no próprio *TensorFlow*⁸, para uso baseado em caracteres.

3) *Aplicação do Treinamento*: Para aplicação do treinamento, optou-se por utilizar inicialmente o a técnica de *Simple*

*RNN*⁹ por meio do *Keras* do próprio *framework*. O uso de Redes Neurais Recorrentes e em específico do *Simple RNN* se justificam por terem facilidade em lidar com informações sequenciais como textos por exemplo [17]. Existem também, outras arquiteturas de Redes Neurais como *GRU (Gated Recurrent Unit)* ou *LSMT (Long Short Term Memory)* por exemplo, que poderão ser exploradas em trabalhos futuros a partir deste estudo.

As *RNNs*, possuem uma arquitetura que permitem ciclos e possuem uma espécie de memória das etapas anteriores, conforme demonstrado na Figura 3. Existe um valor de entrada, que pós processado gera um novo valor de saída, e este por um *loop* retroalimenta a rede passando a informação de uma etapa para outra conservando o seu histórico de evolução mas alterando os pesos conforme cada nova sequência. [17]. O modelo completo da rede pode ser avaliado pela Figura 4.

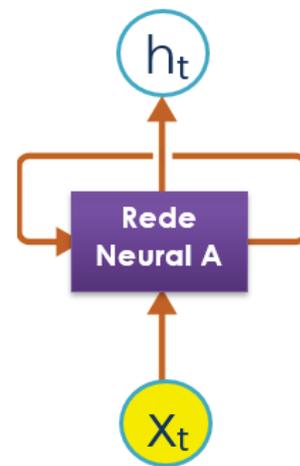


Figura 3. Exemplo de Rede Neural "A", existe uma entrada X_t , e esta gera um valor H_t [17].

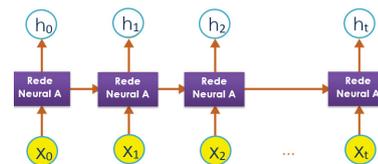


Figura 4. Em uma rede neural, o loop permite que as informações sejam passadas de uma etapa para outra [17].

Para esta rede, foi necessário definir hiperparâmetros buscando foco na performance e no objetivo de uso, pois existem diversas maneiras possíveis de configuração [18]. Foi considerado o valor de 512 para dimensão de entrada, e 1024 neurônios para saída, como descrito na Tabela III. Também foi utilizada uma função de ativação *softmax*, pois ela oferece uma saída percentual em relação a várias entradas sendo comum em problemas de classificação com várias classes.

⁷Biblioteca Markovify. Disponível em: <https://pypi.org/project/markovify/>

⁸Gerador de Texto. Disponível em: https://www.tensorflow.org/text/tutorials/text_generation

⁹Simple RNN com Keras. Disponível em: https://www.tensorflow.org/api_docs/python/tf/keras/layers/SimpleRNN

Com objetivo de auxiliar o treinamento do modelo e evitar o *overfitting*, foi configurado um *dropout* na saída para esta operação em 0.2. A partir destes lotes, criou-se um modelo de vocabulário, e este foi compilado utilizando o otimizador *Adam* para determinar a atualização da rede, escolhido devido a eficiência envolvendo muitos dados e parâmetros [19]. Também foi configurado o *early_stop*, para identificar o melhor ponto de corte nas épocas aplicadas e auxiliar no tempo de processamento monitorando a função de perda (*SparseCategoricalCrossentropy* do próprio *TensorFlow*) que representa a medida de erro do modelo. Embora o *early_stop*, não tenha interrompido nenhum processo, todos os treinamentos foram configurados com 20 épocas devido ao tempo de processamento.

Estes parâmetros foram utilizados em todos os testes de comparação, alterando-se apenas o formato de treino e entrada. Ou seja foi considerado duas opções para processamento do texto e modelagem, sendo uma baseada na palavra completa junto a técnica do *PorterStemmer*, e outra diretamente usando o caractere único. O primeiro teste, considerando a entrada do treino baseado nas palavras com a base completa, mapeou cerca de 531.275 palavras ao todo, sendo 20.384 palavras únicas como mostra o exemplo da Figura 5. Isso somado as configurações explicadas dos hiperparâmetros, resultaram em uma projeção de conclusão para o treino de 26 horas por época aproximadamente.

```
tokenizer.fit_on_texts(corpus)
total_words = len(tokenizer.word_index) + 1
print(tokenizer.word_index)
print(total_words)
{
  "m": 20275, "keyed": 20280, "conclusion": 20281,
  "n": 20285, "headin": 20286, "spreadin": 20287,
  "91": 20292, "what'd": 20292, "mightier": 20293, "crock": 20298,
  "elaborate": 20299, "taut": 20300, "hurting": 20305,
  "feel...": 20306, "bartender": 20307, "311": 20311,
  "exorcism": 20312, "sheepskin": 20313, "remer": 20317,
  "deadman": 20318, "aspects": 20319, "trider": 20323,
  "doorways": 20324, "outworld": 20325, "suit": 20329,
  "combining": 20330, "contaminating": 20331, "n": 20335,
  "sunked": 20336, "optimistic": 20337, "tional": 20341,
  "silince": 20342, "horging": 20347, "shness": 20347,
  "immaculated": 20348, "acheron": 20353, "consent": 20353,
  "azrael": 20354, "go...": 20355, "ight": 20359,
  "vestige": 20360, "enhancing": 20365, "agency": 20365,
  "navigators": 20366, "perpetuated": 20371, "compilation": 20372,
  "breakout": 20376, "necromancy": 20377, "antipope": 20378,
  "clair": 20382, "compulsion": 20383}
20384
```

Figura 5. Exemplo de saída da "tokenização" das palavras.

O modelo baseado em caractere, também com a mesma base completa, embora tenha encontrado 3.331.560 caracteres ao todo, sendo 120 caracteres únicos, foi treinado em 8 horas e 32 minutos com os mesmos hiperparâmetros empregados.

Devido a diferença no custo computacional e tempo do processo entre utilizar o treinamento considerando um modelo de palavras, comparado ao de caracteres, optou-se em realizar o treinamento das demais bases, de artista e de tema restantes apenas com caracteres, descartando a opção de modelagem por palavras. Comparando as três opções de treinamento resultantes (completa, artista e tema), o volume considerado

de músicas na entrada se alteram, bem como tempo de treinamento demonstrados nas Tabelas III e IV, mas a seguir é possível observar o quanto isso de fato altera no resultado predito final.

Tabela III
SIMPLE RNN COMPARATIVOS (HIPER-PARÂMETROS)

	Mod. Palavras	Mod. Caracteres
<i>Emmbeding</i> (D)	512	512
<i>RNN Units</i>	1024	1024
Função de Ativação	<i>softmax</i>	<i>softmax</i>
Épocas	20	20
<i>Dropout</i>	0,2	0,2
Parâmetros Treinados	4.157.584	4.093.305
Tempo de treinamento (horas)	520,3h ¹⁰	8,5h
Função de Perda	-	0,8461

4) *Predições*: Para as predições, foi necessário criar uma função para carregar o modelo e unir o resultado dele ao *input* de entrada com um tamanho definido por caracteres, mas que aceita uma palavra, tema ou até mesmo uma frase para ser a entrada em uma *seed* específica. Com isso, foi possível rodar o modelo inserindo uma frase inicial "how can " como entrada, e o modelo ter o retorno de uma letra de música nova demonstrada na figura 6.

Além da predição realizada com o treino em toda base de 3.250 letras de música, também foram realizadas as predições para os treinos da base de artista *Dark Tranquility* e do tema *Emotions*:

- Predição considerando base de treino **Artist** utilizando a mesma entrada gerando uma nova letra de música que pode ser avaliada pela Figura 7;
- Predição considerando base de treino **Emotions** utilizando também a mesma entrada resultando em uma nova letra de música que pode ser avaliada pela Figura 8.

C. Tecnologias

O equipamento utilizado para análise e construção do modelo, consistiu em um computador pessoal MacBook Pro, 2015, de sistema operacional MAC OS 12.6, 16 Gigas de RAM, com processador Intel Core i7 Dual core a 3,1 GHz.

Neste desenvolvimento, foi adotado como linguagem de programação principal o *Python*, utilizando o *TensorFlow* como *Framework* central para pesquisa, com apoio da ferramenta *Excel* para algumas etapas de pré-processamento das informações. As principais bibliotecas usadas para tratamento e estudo foram:

- *tensorflow.keras.preprocessing.text*: Para o processamento de texto e treinamento do modelo;
- *NLTK*: Conjunto de bibliotecas para Linguagem Natural;
- *Markovify*: Para calcular a probabilidade da ocorrência das palavras;
- *matplotlib*: Para visualizações do modelo;
- *requests*: Biblioteca para facilitar requisições em *http*;

¹⁰Tempo de treinamento simulado com base nas primeiras 2 épocas.

```

how can you be...
i don't give a fuck no

to have held my way out of the car

say you have got so ever what i am

land (walking crowds, there's something you can do our left alone.
started a state for your coffin
something understating feelings in hole? an eclipse.
step outside their honesty, wake up to the never ending starlight
this fever rape that palm the kingdoms go straight into the never)
"
"
need no hope for the living dark

at the ocean of heaven and hell
become the tree, the sun i'm creeping us the war

somebody put something
sucknight silence
landstrial struggle for a 20-knew breeze
memories seem to infectok
all my hearts are caving
headlong down on us
then another way to change our hearts
the storm got my tears
and sometimes we call home
"
"
oh, she cried more finding me
my black letters proclledge
that i'm leaving a perform again

i'll blind ug-instead of infinite life
the cradle of the barther
drown in us, how does the southern child
within the chrontiel with messiah

the sorrow

```

Run time: 3.046218156814575

Figura 6. Resultado da predição com *input* de uma frase-tema.

- BeautifulSoup4: Para etapa de *Web Scraping* das letras em *html*;
- Pandas: Para manipulação e análise de dados;
- Numpy: Para manipulação e transformação dos dados.

II. RESULTADOS E DISCUSSÕES

A. Avaliação dos Resultados

Para este estudo, foi considerado como medida de qualidade o menor resultado de função de perda comparando com o base de dados utilizada, pois ela avalia a distância entre o que foi predito e o valor esperado, representando uma medida de erro do modelo. Embora o modelo de treino e parâmetros sejam os mesmos, são três opções de base onde o tipo de pré-processamento e quantidade de dados na entrada são bem diferentes. Vale ressaltar que a abordagem e o objetivo é auxiliar o compositor com a criatividade em relação a letra da música, o foco principal esta na geração do texto, resultado da predição.

Em relação aos resultados, as três aplicações de base tiveram saídas semelhantes em relação a quantidade de parâmetros treinados como mostra a Tabela IV. Porém, é possível avaliar que o tempo de processamento, teve relação direta com a quantidade de letras e caracteres no banco de dados, mas a

```

how can be

cadry these revanity

i don't ever want to see
i run down to sand fallen
i word to know it

the mistrages of thoughts
on the end of flame
fallen inside and my eye
show me
i am not what that i sing light, the light beicy
to the numbouls atain in mind:

something in belonging of darkness
escangonsed for my mind
the sciting fool -hermine 'ast
a despescen wishing, biter throse things
take he show deepisusflame
shed matters not
walk away
a kent in thoughts
as a vanious barred, never return
disbost strong it
you wnow what i hide
that tell me, it arrows purt too
igain here within
to chay flowed what is wandng
over i could night, forgiveness!
't piry our haunts
now to hatal words
"
"
i this i sagion no pain
a tunnes a nuchor too fall

the things anderth i don't you know

this is it burn our anger
i've deed no pridic
rememorination
so purf for nongestrantly
in the mind - the sketer that me, hert
in this faith reveils of chaos
face again was are
humble, mult is in our eyes
ahide up nightfa

```

Run time: 3.379755735397339

Figura 7. Resultado da predição com *input* de considerando o artista *Dark Tranquility*.

acurácia e a função de perda não. A base *Emotions* obteve um resultado superior de acurácia em 79,8%, e também o menor resultado de erro do modelo em 0,6352 para função de perda com treinamento de 4,8 horas.

B. Contexto das letras geradas, em relação ao banco de dados utilizando WordCloud

Para obter um melhor comparativo visual para o entendimento do contexto das letras geradas a partir da entrada de um banco de dados, utilizar uma *WordCloud* pode facilitar de uma maneira simples a compreensão das principais palavras mais utilizadas [20]. Foi empregado este recurso para comparar



Figura 11. WordCloud gerada a partir das letras ligadas do tema *emotions*.

tipo de treinamento a ser aplicado e na continuidade das abordagens. Pensar na criação do vocabulário amplo para realizar os filtros necessários e ter eficácia no treino, ajudou a evitar que o gerador de texto, entregue letras sem sentido ou palavras que não se conectam, problema comum encontrado durante as pesquisas.

Trabalhar com o *TensorFlow* utilizando os caracteres e com a técnica de Redes Neurais usando *SimpleRNN* mostrou-se eficaz conforme destacado pelas literaturas pesquisadas.

Embora analisando o resultado das *WordCloud*, percebe-se que o contexto principal das palavras não se alterou em relação a quantidade de letras de música em cada uma das três bases, nota-se que é possível utilizar este mesmo conceito com uma base menor porém pré-filtrada para obter melhor performance e resultados. Foi o caso da base segmentada pelo tema *Emotions*, que atingiu a melhor acurácia, com menores resultados na função de perda como mostrou a Tabela IV e conseguiu gerar letras com sentido que podem de fato ajudar na criatividade do músico compositor.

Em relação as demais bases, também tiveram alguns pontos de interesse. A base completa de 3.250 letras, conseguiu um bom resultado na geração de letras, mas pelo fato de possuir diferentes temas relacionados a letra obtida pode ficar confusa. Embora com menos erros de grafia e palavras que não existem, o seu principal ponto positivo, também foi a que teve o maior tempo de treinamento. A base de um artista específico com 168 letras embora um volume pequeno, representou ainda o contexto das letras do estilo, mas acabou gerando letra com maior erros de palavras que não existem.

O uso da IA com Redes Neurais, pode ser aplicado satisfatoriamente nas áreas de entretenimento musical e estimular a criatividade para facilitar o processo de composição, favorecendo a velocidade e continuidade na produção de conteúdo.

Com base neste estudo, espera-se em trabalhos futuros a utilização de outras metodologias de geração de texto, para comparar e explorar o uso utilizando a mesma base de entrada pré-processada por tema. Visando uma melhor construção do vocabulário, atribuição de rimas, melhoria no contexto e custo computacional.

REFERÊNCIAS

- [1] C. R. M. Serra and J. M. de Carvalho, "A arte musical e seu ensino: Desafios e possibilidades do empreendedorismo criativo frente à pandemia de covid-19," *Repositório Institucional UNESP*, 2020. [Online]. Available: <http://hdl.handle.net/11449/202379>
- [2] A. C. Cerqueira, "Viver de música: Empreendedorismo cultural e precarização do trabalho," *Cadernos De Estudos Sociais*, vol. 33, no. 1, pp. 85–107, 2018. [Online]. Available: <https://periodicos.fundaj.gov.br/CAD/article/view/1677>
- [3] A. Triantafyllaki and V. Rowe, "Employing technology in creative music making: Case studies of classroom applications," 2019. [Online]. Available: <https://doi.org/10.34624/musichildren.v0i0.1021>
- [4] I. B. Gorbunova and S. V. Chibirev, "Modeling the process of musical creativity in musical instrument digital interface format," *Revista de Ciencias Humanas y Sociales*, vol. 35, no. 22, pp. 392–409, 2019. [Online]. Available: <https://dialnet.unirioja.es/descarga/articulo/8307925.pdf>
- [5] —, "Mathematical modeling of musical creative process," *2019 3rd International Conference on Art Design, Language, and Humanities*, vol. 3, 2019. [Online]. Available: <https://www.elibrary.ru/item.asp?id=41353901>
- [6] D. M. Thomas and S. Mathur, "Data analysis by web scraping using python," *2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA)*, 2019. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/8822022>
- [7] E. O. Urtaza, "Identity, cosmopolitanism and education in extreme metal bands: The case of finland," *Sociedad Española de Historia de la Educación - HME*, vol. 57, no. 12, pp. 271–302, 2020. [Online]. Available: <https://repositorio.unesp.br/handle/11449/202379>
- [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*, p. 1919–1924, 2015. [Online]. Available: <https://aclanthology.org/D15-1221.pdf>
- [9] H. Liu and Q. Y. an William Yang Wang, "Towards explainable nlp: A generative explanation framework for text classification," 2019. [Online]. Available: <https://doi.org/10.48550/arXiv.1811.00196>
- [10] A. Saeed, S. Ilic, and E. Zangerle, "Creative gans for generating poems, lyrics, and metaphors," 2019. [Online]. Available: <https://doi.org/10.48550/arXiv.1909.09534>
- [11] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery in databases," *AI Magazine*, vol. 17, no. 3, 1996. [Online]. Available: <https://doi.org/10.1609/aimag.v17i3.1230>
- [12] D. N. Wilke, P. W. Cleary, and N. Govender, "From discrete element simulation data to process insights," *EPJ Web of Conferences*, 2021. [Online]. Available: <https://doi.org/10.1051/epjconf/202124915001>
- [13] K. Faceli, A. Lorena, J. Gama, and T. Almeida, *Inteligência Artificial - Uma Abordagem de Aprendizado de Máquina*. LTC, 2011, vol. 1.
- [14] R. Rana, "Heavy metal: A musical subculture and literature," Ph.D. dissertation, Kirtipur, Kathmandu, Nepal, 2021.
- [15] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," 2021. [Online]. Available: <https://doi.org/10.48550/arXiv.2107.13586>
- [16] A. Ankan and A. Panda, *PROC. OF THE 14th PYTHON IN SCIENCE CONF.*, 2015. [Online]. Available: <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.986.409&rep=rep1&type=pdf>
- [17] A. Ferrario and M. Naegelin, "The art of natural language processing: Classical, modern and contemporary approaches to text document classification," 2020. [Online]. Available: <https://ssrn.com/abstract=3547887>
- [18] T. Mitchell, *Machine Learning*. McGraw-Hill Science/Engineering/Math, 1997, vol. 1.
- [19] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Published as a conference paper at the 3rd International Conference for Learning Representations*, 2015. [Online]. Available: <https://doi.org/10.48550/arXiv.1412.6980>
- [20] F. Heimerl, S. Lohmann, S. Lange, and T. Ertl, "Word cloud explorer: Text analytics based on word clouds," *Hawaii International Conference on System Sciences*, 2014.