UNIVERSIDADE FEDERAL DE GOIÁS / INSTITUTO DE INFORMÁTICA

Geração Musical Para Games

Uma Abordagem Tex-to-Music Condicionada à Emoção

Luiz Fernando de Araújo Vidal

UFG

UNIVERSIDADE
FEDERAL DE GOIÁS

UNIVERSIDADE FEDERAL DE GOIÁS (UFG)
INSTITUTO DE INFORMÁTICA (INF)

LUIZ FERNANDO DE ARAÚJO VIDAL

GERAÇÃO MUSICAL PARA GAMES

Uma Abordagem Tex-to-Music Condicionada à Emoção

Goiânia
2024

UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR VERSÕES ELETRÔNICAS DE TRABALHO DE CONCLUSÃO DE CURSO DE GRADUAÇÃO NO REPOSITÓRIO INSTITUCIONAL DA UFG

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio do Repositório Institucional (RI/UFG), regulamentado pela Resolução CEPEC no 1240/2014, sem ressarcimento dos direitos autorais, de acordo com a Lei no 9.610/98, o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou download, a título de divulgação da produção científica brasileira, a partir desta data.

O conteúdo dos Trabalhos de Conclusão dos Cursos de Graduação disponibilizado no RI/UFG é de responsabilidade exclusiva dos autores. Ao encaminhar(em) o produto final, o(s) autor(a)(es)(as) e o(a) orientador(a) firmam o compromisso de que o trabalho não contém nenhuma violação de quaisquer direitos autorais ou outro direito de terceiros.

1. Identificação do Trabalho de Conclusão de Curso de Graduação (TCCG)

Nome(s) completo(s) do(a)(s) autor(a)(es)(as): **LUIZ FERNANDO DE ARAÚJO VIDAL**

Título do trabalho:

GERAÇÃO MUSICAL PARA GAMES

Uma Abordagem Text-to-Music Condicionada à Emoção

2. Informações de acesso ao documento (este campo deve ser preenchido pelo orientador) Concorda com a liberação total do documento [**X**] SIM [ ] NÃO[1]

[1] Neste caso o documento será embargado por até um ano a partir da data de defesa. Após esse período, a possível disponibilização ocorrerá apenas mediante: a) consulta ao(à)(s) autor(a)(es)(as) e ao(à) orientador(a); b) novo Termo de Ciência e de Autorização (TECA) assinado e inserido no arquivo do TCCG. O documento não será disponibilizado durante o período de embargo.

Casos de embargo:

- Solicitação de registro de patente;

- Submissão de artigo em revista científica;

- Publicação como capítulo de livro.

Obs.: Este termo deve ser assinado no SEI pelo orientador e pelo autor.

Documento assinado eletronicamente por **Luiz Fernando De Araújo Vidal**, **Discente**, em 16/02/2024, às 15:29, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Fernando Marques Federson**, **Professor do Magistério Superior**, em 12/09/2024, às 11:01, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade deste documento pode ser conferida no site https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **4386327** e o código CRC **65942BA3**.

Referência: Processo nº 23070.008392/2024-11 SEI nº 4386327

LUIZ FERNANDO DE ARAÚJO VIDAL

GERAÇÃO MUSICAL PARA GAMES

Uma Abordagem Tex-to-Music Condicionada à Emoção

Relatório final de Trabalho de Conclusão de Curso, apresentado à Universidade Federal de Goiás, como parte das exigências para a obtenção do título de Bacharel em Inteligência Artificial.
Orientador: Prof. Dr. Fernando Marques Federson

Goiânia
2024

Ficha de identificação da obra elaborada pelo autor, através do
Programa de Geração Automática do Sistema de Bibliotecas da UFG.

VIDAL, LUIZ FERNANDO DE ARAÚJO
GERAÇÃO MUSICAL PARA GAMES [manuscrito] : Uma Abordagem Text-to-Music Condicionada à Emoção / LUIZ FERNANDO DE ARAÚJO VIDAL. - 2024.
81 f.

Orientador: Prof. Dr. FERNANDO MARQUES FEDERSON.
Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Goiás, Instituto de Informática (INF), Inteligência Artificial, Goiânia, 2024.

1. inteligência artificial. 2. geração musical. 3. games. I. FEDERSON, FERNANDO MARQUES, orient. II. Título.

CDU 004

LUIZ FERNANDO DE ARAÚJO VIDAL

GERAÇÃO MUSICAL PARA GAMES

Uma Abordagem Tex-to-Music Condicionada à Emoção

Relatório final de Trabalho de Conclusão de Curso, apresentado à Universidade Federal de Goiás, como parte das exigências para a obtenção do título de Bacharel em Inteligência Artificial.

Data da Aprovação: 08 de fevereiro de 2024.

Prof. Dr. Fernando Marques Federson
Orientador (INF-UFG)

Prof. Dr. Aldo André Díaz Salazar
Coordenador de TCC do BIA (INF-UFG)

Prof. Dr. Vinícius Sebba Patto
Coordenador do BIA (INF-UFG)

Prof. Dr. Arlindo Rodrigues Galvão Filho
(INF-UFG)

LUIZ FERNANDO DE ARAÚJO VIDAL

GERAÇÃO MUSICAL PARA GAMES

Uma Abordagem Text-to-Music Condicionada à Emoção

RESUMO

Este Relatório de Conclusão de Curso tem como objetivo reunir os resultados da minha jornada para me tornar um especialista em **Music Information Retrieval (Music Generation)**. Uma ilustração e sua narrativa descrevem os períodos de trabalho. Os Apêndices contêm os Termos de Aceite de Entrega e os resultados obtidos durante cada período de trabalho.

Palavras-chave: inteligência artificial, geração musical, games.

ABSTRACT

This Course Completion Report aims to bring together the results of my journey to become an expert in **Music Information Retrieval (Music Generation)**. An illustration and its narrative describe the work periods. The Appendices contain the Delivery Acceptance Terms and the results obtained during each work period.

Keywords: artificial intelligence, music generation, games.

Goiânia

2024

Minha Jornada

Luiz Fernando de Araújo Vidal

Especialista em: Music Information Retrieval (Music Generation)

MINHA JORNADA

Nome:

Especialidade:

Objetivo deste documento

Durante o processo da disciplina Residência em IA[1], foram gerados diversos resultados na construção da minha especialização. A cada semana, um conjunto de resultados foi formalizado por um Termo de Aceite de Entrega e avaliado por uma banca, considerando o planejado e o realizado para o período. Este documento tem como objetivo descrever esses resultados obtidos, fazendo referência aos Termos de Aceite de Entrega e seus documentos associados.

Minha Jornada

Toda a minha jornada teve início antes da primeira semana de entregas, pois o tema e área que foram apresentados por mim na **Semana 1** foram frutos de uma aula muito interessante sobre IA Aplicada à Música, ou *Music Information Retrieval* (MIR), ministrada pelo Prof. Dr. Arlindo Galvão na Disciplina de Processamento de Áudio e Voz (PAV). Nesse sentido, uma chave virou dentro de mim e a Residência em IA foi o momento ideal para colocar em prática as ideias que foram construídas ao longo do tempo dentro de mim. A partir disso, **a Semana 1** foi o momento de apresentar o tema “Abordagem Text-To-Audio, Condicionada à Emoção, para Geração de Música para Games”, bem como suas motivações, e um estudo geral sobre as diversas tarefas que compõem a área de MIR. Além disso, foi criado um repositório de artigos e links, separados por tópicos, com o objetivo de centralizar os materiais importantes para os meus primeiros passos na Geração Musical.

Com o tema já definido, a **Semana 2** foi completamente dedicada a estudar as emoções do ponto de vista da Psicologia, mas também a importância dessas emoções no contexto da música e dos games. Nesse estudo, além da Teoria das Emoções Universais de

[1] Dez semanas, entre setembro de 2023 e janeiro de 2024.

Paul Ekman, que propõe a existência de 7 emoções básicas universais inatas e biologicamente enraizadas, explorei também as ideias de Lisa Feldman Barrett. A abordagem de Barrett, que enfatiza o construtivismo emocional e a influência dos fatores culturais e ambientais na experiência emocional, forneceu um contraponto valioso e uma perspectiva mais ampla. A combinação dessas teorias permitiu uma visão mais completa e complexa sobre as emoções, o que foi fundamental para escolher as emoções mais apropriadas para condicionar a geração musical. Os detalhes e referências dos estudos desenvolvidos durante essas duas semanas podem ser encontrados no **Apêndice 1**.

Após fazer essa imersão teórica acerca das emoções, e entender sua importância no contexto musical e dos games, foi possível direcionar os meus esforços para colocar em prática a geração musical condicionada a emoção para games. Desse modo, na **Semana 3**, foi feita uma seleção de conjuntos de dados (datasets) para geração musical que possuíssem algum tipo de anotação emocional. Foram selecionados 3 datasets sendo que um deles, o YM2413-MDB que foi apresentado no *International Society for Music Information Retrieval Conference* (ISMIR) de 2022 , se encaixava perfeitamente na minha aplicação: um dataset de músicas de games 8 bits do console *Super Nintendo Entertainment System* (SNES). Além disso, com um primeiro dataset em mãos, o próximo passo foi escolher um modelo *text-to-music* ideal para o desenvolvimento do trabalho. Com base no critério de disponibilidade de código e facilidade de realizar *fine-tunings*, o MusicGen, da Meta, foi o candidato perfeito. Por conta de ser um modelo bem utilizado pela comunidade, o *pipeline* de *fine-tuning* dele estava disponível e bem documentado na plataforma Replicate. Além de me auxiliar a preparar os meus primeiros experimentos, a plataforma também me forneceu a informação de que não seria necessário muitas músicas para realizar o *fine-tuning* do MusicGen, pois o modelo consegue condicionar sua geração com poucos exemplos musicais e um *prompt* simples. Para a execução do *pipeline* da Replicate, decidi utilizar a API com alocação de GPU paga, buscando uma performance otimizada para os experimentos. Paralelamente, enfrentei o desafio de executar o *pipeline* da Stable Audio Tools no Google Colab, utilizando os recursos computacionais gratuitos oferecidos por esse ambiente. Sabendo que esses recursos seriam mais escassos e limitados, o objetivo tornou-se otimizar ao máximo o pipeline para garantir sua execução de forma eficaz. Enquanto na Replicate o condicionamento seria realizado por meio de *prompts* textuais, no

Stable Audio Tools, a abordagem seria diferente, permitindo a inserção direta da emoção discretizada durante o *fine-tuning*. Os detalhes e as referências para essa importante semana podem ser encontrados no **Apêndice 2**.

Durante a **Semana 4**, foi possível desenvolver a primeira versão do meu dataset de músicas de games modernos que foi montado com base nas emoções “feliz”, “triste” e “calmo”, onde cada música possuia um arquivo de texto contendo um *prompt* de condicionante para o modelo como, por exemplo, *“happy, cheerful, game music”*. Para essa primeira versão, foram obtidas 25 músicas felizes, 19 tristes e 30 calmas de playlists do Youtube que faziam referência à emoção em seus títulos, logo, foi uma maneira de realizar a auto-anotação dessas músicas. Vale destacar que a emoção “calmo” não faz parte da teoria de Paul Ekman (Teoria das Emoções Universais), mas foi uma emoção utilizada no dataset YM2413-MDB. Além disso, nesta semana, realizei as primeiras tentativas de *fine-tuning* com o *pipeline* da Stable Audio Tools no dataset de games 8 bits. No entanto, mesmo com uma pequena amostra de áudios, reduções na taxa de amostragem e no tamanho das amostras de áudio, enfrentei dificuldades significativas, pois o Google Colab não dispunha de recursos suficientes para suprir as necessidades do *fine-tuning*. Isso deixou bem claro para mim que, talvez, fosse necessário ir mais a fundo no processo de otimização do *fine-tuning*. Paralelamente, enfrentei problemas com o *pipeline* da Replicate. Durante a tentativa de utilização da API, deparei-me com um problema técnico que me levou a recorrer à comunidade em busca de uma possível solução, processo esse, que me fez pensar em como uma comunidade ativa é importante nesses momentos, pois foi graças ao suporte de um usuário da plataforma que eu consegui dar prosseguimento aos meus trabalhos.

A partir disso, na **Semana 5**, o foco foi gerar a versão 2 do dataset ao inserir 13 músicas da emoção “medo” no dataset, totalizando agora 77 músicas. Desse modo, com o conjunto de dados atualizado em mãos, avancei no projeto realizando a validação do *pipeline* da Replicate para o *fine-tuning* do modelo MusicGen Medium. Este processo foi particularmente notável, pois foi executado usando uma infraestrutura robusta de 8 GPUs A40, acompanhadas de 600 GB de RAM. Essa configuração de hardware proporcionou um ambiente de computação de alta performance, crucial para o sucesso do treinamento do modelo que gerou resultados muito interessantes. Um aspecto importante a destacar é o custo associado a este processo, visto que o *fine-tuning*, com esta infraestrutura de ponta,

teve um custo de 12 dólares. Por outro lado, além do progresso com o *pipeline* da Replicate, continuei o processo de otimização do *pipeline* da Stable Audio Tools. Dada a limitação de recursos no Google Colab, foquei em mudanças na precisão do modelo, buscando reduzir o uso de memória e viabilizar o *fine-tuning*. Esta abordagem incluiu a alteração de parâmetros do modelo para versões que demandam menos recursos computacionais. No entanto, apesar desses esforços, a otimização não resolveu completamente o problema, pois as limitações de memória e processamento ainda eram significativas, impedindo o sucesso do *fine-tuning* no ambiente do Google Colab. Com isso, essa experiência me levou à conclusão de que é necessário contar com uma infraestrutura mais robusta para trabalhar eficientemente com o modelo MusicGen. A demanda por recursos computacionais elevados fez com que o foco se voltasse exclusivamente para o *pipeline* do Replicate e sua abordagem textual, logo, esta decisão refletiu em uma adaptação estratégica às condições de hardware disponíveis, priorizando o caminho que oferecia maior viabilidade técnica e econômica. Os detalhes completos sobre esses ajustes, decisões e seus impactos no projeto podem ser encontrados no **Apêndice 3**.

Para a **Semana 6**, o principal objetivo foi criar uma nova versão do dataset com a adição de novas músicas para certas emoções e a mudança do formato de áudio mp3 para o wav, pois a compressão realizada no mp3 causa a perda de muitas informações a nível espectral, o que poderia atrapalhar nos *fine-tunings*. Além da mudança de formato, na versão 3 do dataset foram adicionadas mais 40 músicas para a emoção “calmo” e 19 músicas para a emoção “sad”, totalizando um aumento de um pouco mais de 1 hora para essas emoções. O dataset passou a ter um total de 6 horas e 23 minutos com 146 músicas. Além disso, os resultados obtidos no primeiro *fine-tuning* do MusicGen Medium foram avaliados por meio da anotação humana com base nos critérios de correção da emoção, ou seja, se está “certo” ou “errado” e se a qualidade do áudio gerado é “boa”, “média” ou “ruim”. Partindo disso, as avaliações foram bem interessantes, pois para o primeiro critério a emoção “triste” não teve um desempenho satisfatório em relação às outras emoções, mas no geral o modelo conseguiu gerar músicas que fossem coerentes com os seus *prompts*. Por outro lado, no critério de qualidade, as avaliações não foram tão uniformes como para a emoção, embora nenhuma música tenha sido classificada como “ruim”. Durante essa semana, também, tentei realizar dois novos *fine-tunings* usando o MusicGen

Stereo-Medium, sendo um com o dataset criado por mim e outro com o dataset de games 8 bits, mas o *pipeline* não estava atualizado para que isso fosse possível e os resultados foram problemáticos. Todos os detalhes sobre a nova versão do dataset e as avaliações de resultados do primeiro fine-tuning podem ser visualizados no **Apêndice 4**.

Com base no que foi observado durante a **Semana 6**, para a **Semana 7**, considerei ser importante avaliar quais melhorias poderiam ser feitas nos prompts ou quais recursos do *pipeline* de *fine-tuning* poderiam ser interessantes utilizar dessa vez. Nesse sentido, com essas mudanças feitas, criei uma nova versão do dataset que seria a união dos dois datasets já trabalhados anteriormente, assim, foi possível obter um único conjunto de dados que possui músicas de games modernos e antigos, totalizando 618 músicas e cerca de 50 horas de áudio. Logo após isso, foi feito o *fine-tuning* com o *pipeline* agora já atualizado do MusicGen Stereo-Medium e obtive resultados iniciais bem interessantes.

Ainda na **Semana 7**, iniciei a implementação de um minigame baseado na franquia Castlevania para que, no futuro, uma música gerada por mim fosse inserida no game. Toda a implementação inicial do código e dos assets foi feita com o Chat-GPT e Dall-E (**Apêndice 5**).

Durante a **Semana 8**, além de realizar melhorias na jogabilidade do minigame e na arte, me dediquei a expandir meu conhecimento e explorar novas possibilidades no campo da geração musical. Nesse contexto, mergulhei em um estudo detalhado de uma playlist sobre geração musical, buscando inspirações e ideias novas. Foi nesse processo de pesquisa e descoberta que me deparei com um modelo particularmente interessante chamado RAVE (*Realtime Audio Variational autoEncoder*). O RAVE apresentou-se como uma ferramenta promissora, com um potencial de gerar música a partir de dados não convencionais, como por exemplo, sinais de EEG (Eletroencefalograma). Essa característica abriu um leque de possibilidades para experimentações futuras, onde poderia explorar a interação entre estados cerebrais e a criação musical. Uma frente empolgante que mistura música, tecnologia e neurociência. Animado com essa descoberta, considerei a possibilidade de integrar o RAVE em projetos futuros. Esta linha de pesquisa representou um potencial caminho inovador para a minha jornada em *Music Information Retrieval* (MIR), adicionando uma nova dimensão aos meus experimentos com geração musical. As análises dos resultados do *fine-tuning* realizado na **Semana 7**, feitas por meio da anotação humana,

também prosseguiram nesta semana. Nesse sentido, enquanto os resultados para músicas de games modernos foram muito bons, os para músicas de games 8 bits não atingiram níveis satisfatórios, principalmente em termos de qualidade sonora. Essas avaliações, juntamente com a descoberta do modelo RAVE e seus potenciais usos, estão documentadas em detalhes no **Apêndice 5**.

Chegando quase ao fim da minha jornada, na **Semana 9**, me dediquei intensamente ao estudo e montagem do *pipeline* de treino do modelo RAVE, pois precisava compreender a fundo as capacidades técnicas do modelo e prepará-lo para o treinamento. Paralelamente, explorei artigos que relacionavam geração musical e neurociência, buscando validar a ideia de gerar músicas condicionadas a emoções a partir de sinais biológicos. Encontrei 3 artigos que me trouxeram certa tranquilidade sobre a viabilidade da ideia. Na **Semana 10**, após diversas iterações, concluí o treinamento do RAVE. Apesar disso, enfrentei dificuldades com a função de exportação do modelo, que não estava funcional no repositório oficial, limitando a utilização prática do modelo treinado. Na outra linha de desenvolvimento, realizei um novo *fine-tuning* com o modelo Stereo-Melody do MusicGen, focando em melhorar os resultados para as músicas de games 8 bits. Após avaliar os resultados obtidos, planejei uma reavaliação para a semana seguinte com um grupo mais amplo de pessoas. Como última atividade do período, finalizei a implementação do minigame com melhorias, novas funcionalidades e um “chefe desafiador”. A parte mais interessante foi gerar a música do game utilizando a música *“Crystal Teardrops”* da trilha sonora de *Castlevania Symphony of The Night* como base melódica e o *prompt “tense, dark, frightening, 8bit game music”*.

Na **Semana 11**, procedi com a reavaliação dos resultados do *fine-tuning* do MusicGen, realizada por meio da anotação humana. As avaliações mostraram uma melhoria notável na correção emocional e na qualidade sonora. Esse progresso ressaltou a eficácia das otimizações nos *prompts* das músicas e no processo de *fine-tuning* realizados. Todos os detalhes sobre estas etapas, incluindo os desafios com o modelo RAVE, a conclusão do minigame e os resultados do fine-tuning do MusicGen, estão documentados no **Apêndice 6**.

Diante disso, posso afirmar com muita certeza que a Disciplina Residência em IA foi um marco muito importante na minha vida dentro do curso, pois a liberdade que é

proporcionada a nós, alunos, para explorar as áreas de aplicação da IA, gera uma sensação de acolhimento e compreensão, no caso, não há pressão para escolher áreas que estão em alta ou que são tendência. No meu caso, entrei na Residência já com a mentalidade de que estudaria MIR (*Music Information Retrieval*), com foco nos modelos generativos de música, mas não imaginava que gostaria tanto desse assunto. Por fim, gostaria de agradecer todo o suporte oferecido pelos professores da Disciplina, posso dizer que finalmente me encontrei dentro do curso. O processo da Disciplina Residência em IA tornou-se muito prazeroso e animador para mim, de tal forma que me faz considerar a possibilidade de seguir meus estudos em uma Pós-Graduação, o que era uma realidade muito distante para mim.

APÊNDICE 1

Termo de Aceite de Entrega 1

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 19 de out. de 2023

Participantes da Entrega [matriculados em Residência em IA]:

Luiz Fernando de Araújo Vidal

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Requisitos cumpridos:

- Definição do tema do TCC: Abordagem text-to-audio, condicionada à emoção, para geração de música para games
- Classificação do projeto de acordo com os tópicos da CSCI 2023:
  - Escopo: Artificial Intelligence
  - Tópicos: Natural Language Processing, Neural Networks and Applications e Signal Processing

Produtos gerados:

Os produtos gerados para essa entrega envolvem um estudo sobre a Music Information Retrieval (MIR) com o objetivo de compreender as tasks que existem dentro do cenário de IA aplicada à música, alguns links de motivação para a escolha do meu tema de TCC e também o início da construção de um repositório de artigos que irão auxiliar no processo de estudo de geração musical.

Estudo sobre MIR: Estudo sobre MIR

Motivações para a elaboração do projeto:

- Game Music Composer: Understanding Rates and Costs
- Understanding How Much an Indie Game Music Composer Costs — Ninichi
- How much does music for a game cost?

Referências para Geração Musical: Repositorio de artigos-ResIA

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Planejamento para a próxima entrega:

- Continuar a busca por artigos para o repositório
- Iniciar o estudo sobre o campo das emoções
- Entender a importância da emoção nos games e nas músicas

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

LUANA GUEDES BARROS MARTINS: Go!

ESTUDO SOBRE MIR

Basicamente, a Music Information Retrieval (MIR) é uma ciência interdisciplinar dedicada à recuperação de informações a partir da música. Ela envolve diversas áreas de conhecimento, incluindo musicologia acadêmica, psicoacústica, psicologia, processamento de sinais, aprendizado de máquina, e inteligência computacional. Nesse sentido, abaixo estão listadas algumas das tasks que, atualmente, são desenvolvidas a partir desse campo de conhecimento:

- Classificação
  - Gênero musical;
  - Artista;
  - Sentimento;
  - Instrumento;

- Recomendação

- OMR (Reconhecimento Óptico de Música)
  - OMR envolve a conversão de uma partitura física ou impressa em uma representação digital, como um arquivo MusicXML ou MIDI.
  - Utiliza tecnologias de visão computacional e processamento de imagem para identificar e interpretar símbolos musicais em uma página

- Source separation;
  - Separação de sons de instrumentos distintos;
  - Similar a diarização de locutor;

- Transcrição musical automática (AMT);
  - Tem o objetivo de gerar alguma representação musical, que pode ser simbólica ou notação formal
  - Partituras, tablaturas, XML, MID
  - Detecção a nível de frame ou de nota;
  - Sons polifônicos ou monofônicos.

- Geração musical.

- Consiste na geração do música a partir de textos, trechos de músicas e representações simbólicas;
- Controlar geração, qual gênero? Qual velocidade? Quais notas? etc.
- Mais desafiador que geração de fala, pois requer o entendimento de longas sequências;
- Ser humano consegue captar facilmente sons em desarmonia, o que é um desafio para a geração de música.

Referências:

- Music Information Retrieval: Recent Developments and Applications
- Music Information Retrieval - an overview | ScienceDirect Topics

Termo de Aceite de Entrega 2

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 26 de out. de 2023

Participantes da Entrega [matriculados em Residência em IA]:

Luiz Fernando de Araújo Vidal

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Entrega:

- Continuação da busca por artigos para o repositório
- Início dos estudos sobre o campo das emoções
- Compreensão da importância da emoção nos games e nas músicas

Link do repositório de links e artigos: Repositorio de links e artigos-ResIA
Link para o estudo: Estudo sobre emoções_GATE_261023

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Próximos passos:

- **Buscar maneiras de condicionar a geração musical à emoção**
- **Pesquisar datasets de músicas que possuam as emoções anotadas**
- **Analisar a possibilidade de construir um dataset próprio**
- **Encontrar o modelo generativo text-to-audio ideal para o TCC**

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

LUANA GUEDES BARROS MARTINS: Go!

ESTUDO SOBRE O CAMPO DAS EMOÇÕES

Primeiro momento do estudo:

Emoções sob a perspectiva da psicologia:
O intuito deste primeiro momento de estudo é entender, do ponto de vista da psicologia, como ocorre a compreensão e formação das emoções humanas. Nesse sentido, tal assunto é considerado muito complexo e importante até os dias de hoje no mundo acadêmico, entretanto, duas teorias ganharam notoriedade para explicar, com abordagens diferentes, o tópico em questão. Desse modo, será feito um paralelo entre a Teoria das Emoções Universais de Paul Ekman e a Teoria da Construção das Emoções de Lisa Feldman com o objetivo de extrair informações interessantes para o desenvolvimento do TCC.

Teoria das Emoções Universais de Paul Ekman:
- **Emoções Básicas**: Ekman propôs que existem seis emoções básicas que são universais entre todas as culturas humanas: alegria, tristeza, medo, raiva, surpresa e repulsa. Ele mais tarde expandiu essa lista para incluir outras emoções como desprezo.
- **Universais e Biologicamente Enraizadas:** Ekman acreditava que essas emoções básicas são universais e têm raízes biológicas. Isso significa que elas são inatas e não são aprendidas culturalmente. Desse modo, tal teoria tem sido aplicada em diversas áreas como a psicologia clínica, forense e no desenvolvimento de inteligência artificial. No campo da inteligência artificial, especialmente em tarefas como análise de sentimento em texto ou reconhecimento de emoções na voz, geralmente são anotadas nos datasets as 7 emoções básicas teorizadas por Ekman. Isso indica uma predominância dessa teoria no campo da computação e da IA aplicada..
- **Expressões Faciais**: Uma das principais contribuições de Ekman foi sua pesquisa sobre expressões faciais. Ele argumentou que cada emoção básica tem uma expressão facial específica associada que pode ser reconhecida em todas as culturas. Ele desenvolveu o Facial Action Coding System (FACS) para categorizar essas expressões.

Teoria da Construção das Emoções de Lisa Feldman Barrett:
- **Construção das Emoções**: Barrett propõe que as emoções não são estados biologicamente universais que têm correspondências claras e distintas no cérebro. Em vez disso, elas são construídas por sistemas cerebrais mais gerais que trabalham juntos, e não há circuitos cerebrais dedicados exclusivamente a emoções específicas.
- **Variação Cultural**: Ao contrário da perspectiva de Ekman, Barrett argumenta que a maneira como as pessoas categorizam e percebem emoções pode variar significativamente

entre diferentes culturas. As categorias emocionais que temos (como "raiva" ou "felicidade") são construções culturais e podem não ter equivalentes exatos em outras culturas.

- **Contexto é Crucial**: Para Barrett, o contexto desempenha um papel crucial na forma como interpretamos e nomeamos nossas experiências emocionais. Ela argumenta que o cérebro usa informações do contexto presente, juntamente com nossas experiências passadas, para construir nossas experiências emocionais.

Em resumo, enquanto Ekman vê emoções como universais e biologicamente enraizadas com expressões faciais específicas associadas, Lisa Feldman vê emoções como construções que são moldadas pelo contexto e pela cultura. Além disso, a teoria apresentada pela Lisa Feldman introduz uma complexidade muito grande para a análise e compreensão das emoções, pois cada indivíduo se tornaria singular ao se tratar das emoções, em contrapartida, Ekman permite que o estudo das emoções seja mais generalista para que haja um padrão de estudo emocional mesmo quando indivíduos diferentes são estudados. Portanto, pensando a nível de pesquisa, Ekman possui uma teoria mais aplicável.

Conclusão:
Nesse sentido, a partir desse paralelo foi possível perceber que a abordagem mais coerente com o desenvolvimento do meu TCC seria a de Paul Ekman, pois as emoções universais podem ser utilizadas para descrever um dataset de músicas para games, logo, seria possível criar um elo entre as estruturas músicas e as 7 emoções básicas. Por fim, essas informações poderiam ser utilizadas para condicionar a geração musical do modelo.

Segundo momento do estudo:

Neste segundo momento, irei sintetizar a compreensão que eu tive sobre a importância da música como uma ferramenta de evocar emoções e também sobre o papel das emoções em criar uma experiência única para os jogadores.

Importância da música e emoções nos Games:

Os jogos evoluíram de simples entretenimentos pixelados para formas de arte profundamente envolventes e emocionais. O elemento principal dessa transformação é como a música e a emoção trabalham em sintonia para aprimorar a experiência do jogador.

Nesse sentido, a música, por si só, sempre foi uma expressão profunda das emoções humanas. Desde tempos antigos, ela tem sido usada para comunicar sentimentos, contar histórias e evocar reações emocionais nas pessoas. Uma canção triste pode revisitar memórias dolorosas, enquanto outra pode despertar

sentimentos felizes e nostálgicos. Com isso, o poder único da música de ressoar e amplificar emoções é o que a torna tão importante nos jogos, pois o simples fato de um game possuir uma trilha sonora marcante torna-o memorável por anos.

Dentro do universo dos videogames, a música serve como uma ponte entre a narrativa e o jogador, intensificando e moldando as emoções apresentadas. Em cenários críticos, como a perda de um personagem ou uma batalha intensa, a trilha sonora amplifica a emoção do momento. Jogos de terror, como os da franquia Resident Evil, sabem explorar muito bem o uso do recurso musical ao seu favor, pois, principalmente nos títulos mais clássicos, o jogador, a qualquer momento, pode entrar em uma situação crítica e tensa, mas, ao conseguir alcançar a famosa “save room”, o clima do jogo muda completamente com a introdução de uma música calma e tranquila, o que passa uma sensação de paz e segurança para o jogador. Desse modo, a música é um elemento extremamente importante para que a experiência proposta pelo game, seja ela gerar medo, tensão ou felicidade, seja a mais fiel possível.

A importância das emoções nos jogos também é evidenciada por inovações como o "affective game loop". Esta técnica moderna permite que os jogos capturem e respondam às emoções dos jogadores em tempo real. Isso possibilita uma adaptação do jogo à experiência do jogador, tornando-a dinâmica e personalizada. Por exemplo, diante de um jogador frustrado, o jogo pode sugerir dicas; se detectar relaxamento, pode introduzir um desafio surpreendente.

Com isso, tanto a música quanto a narrativa são instrumentos poderosos para estabelecer conexões emocionais. Os jogadores estão mais engajados quando sentem uma ligação emocional com um personagem ou história, tornando esses momentos particularmente memoráveis. Portanto, isso mostra o potencial que os jogos possuem para se conectar com os jogadores em níveis emocionais profundos. Conforme os videogames evoluem, a integração de música, emoção e design adaptativo se torna ainda mais central para amplificar a experiência do jogador.

Referências:

Estudo de emoções (Psicologia):

1. Universal Emotions | What are Emotions? | Paul Ekman Group
2. Teorias das Emoções - Paul Ekman e Lisa Feldman (com 1 aula)
3. Lisa Feldman Barrett: Você não está à mercê das suas emoções - seu cérebro as cria | TED Talk

4. Paul Ekman, um psicólogo além do seu tempo - Linguagem Corporal

Estudo de emoções nos games e na música:

1. O poder que as músicas de videogames têm sobre nossas emoções - BBC News Brasil
2. Um estudo sobre áudio como elemento imersivo em jogos eletrônicos
3. The importance of music in video games
4. Emotion in Games
5. (PDF) Emotional Gaming.
6. Unlocking Sound: How Music Influences the Gaming Experience
7. Influencing emotions in game design: theories and methods

As referências encontram-se, também, no meu repositório de links e artigos:
Repositorio de links e artigos-ResIA

APÊNDICE 2

Termo de Aceite de Entrega 3

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 9 de nov. de 2023

Participantes da Entrega [matriculados em Residência em IA]:

Luiz Fernando de Araújo Vidal

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

As entregas programadas para o atual gate foram:

- **Pesquisar datasets de músicas que possuam as emoções anotadas**
- **Analisar a possibilidade de construir um dataset próprio**
- **Encontrar o modelo generativo text-to-audio ideal para o TCC**
- **Buscar maneiras de condicionar a geração musical à emoção**

Documento criado para a entrega: Entrega Gate 09/11/2023

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

- **Buscar novidades apresentadas no ISMIR 2023**
- **Preparar o ambiente para o pipeline de fine-tuning do Replicate e iniciar experimentos**
- **Realizar mais testes com o pipeline de fine-tuning da stable audio tools (Stability-AI)**
- **Montar um pequeno dataset com músicas de um jogo de gênero específico para validar no pipeline do Replicate**

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

LUANA GUEDES BARROS MARTINS: Go!

Entrega Gate 09/11/2023

1. **Pesquisar datasets de músicas que possuam as emoções anotadas**
   Foram encontrados alguns datasets interessantes que serão descritos logo abaixo:
   - ISMIR 2022: YM2413-MDB: A Multi-Instrumental FM Video Game Music Dataset with Emotion Annotations
     O artigo descreve o "YM2413-MDB", um dataset de música de video game dos anos 80 com anotações de emoções. O conjunto de dados inclui 669 arquivos de áudio e MIDI de músicas de jogos da Sega e MSX PC usando o YM2413, um gerador de som programável baseado em FM. A música dos jogos coletados é organizada com um subconjunto de 15 instrumentos monofônicos e um instrumento de bateria, convertidos a partir de comandos binários do chip de som YM2413. Cada música foi rotulada com 19 tags de emoções por dois anotadores e validada por três verificadores para obter tags refinadas. Além disso, o artigo fornece modelos de base e resultados para reconhecimento de emoções e geração de música simbólica condicionada por emoções usando o YM2413-MDB.
   - [2108.01374] EMOPIA: A Multi-Modal Pop Piano Dataset For Emotion Recognition and Emotion-based Music Generation
     O dataset multimodal (áudio e MIDI) descrito no artigo é focado na emoção percebida na música de piano pop. Nesse sentido, ele contém 1087 clipes de música extraídos de 387 músicas, com rótulos de emoção em nível de clipe, anotados por quatro anotadores. Os clipes são segmentados de forma que podem ser usados para análise em nível de música também.
   - DEAM: MediaEval Database for Emotional Analysis in Music
     O dataset DEAM (MediaEval Database for Emotional Analysis in Music) inclui mais de 1800 músicas com anotações emocionais. Este dataset possui dados dos anos de 2013 a 2015 do desafio "Emotion in Music" da MediaEval, além de rótulos brutos. As anotações emocionais foram coletadas com o objetivo de detectar as emoções expressas pela música e pelos músicos.
     O DEAM contém música isenta de copyrights de várias fontes. O conjunto de dados é composto pelo conjunto de desenvolvimento de 2014 (744 músicas), o conjunto de avaliação de 2014 (1000 músicas) e o conjunto de avaliação de 2015 (58 músicas). Nesse sentido, trechos de 45 segundos foram extraídos de pontos aleatórios (distribuição uniforme) dentro das músicas e codificados para ter a mesma frequência de amostragem de 44100Hz.

2. **Analisar a possibilidade de construir um dataset próprio**
   Após pesquisar mais sobre o fine-tuning do modelo escolhido (MusicGen), descobri que em alguns pipelines incluem a auto-anotação das músicas utilizando uma ferramenta chamada Essentia. Logo, com essa biblioteca é possível gerar labels de mood (que incluem as emoções), gênero que se encaixa na música, beat e entre outras informações possíveis de se extrair do áudio.

Além disso, outra informação pertinente é que o MusicGen não precisa de muitos dados para o seu fine-tuning, visto que ele consegue aprender um estilo musical com uma amostra de 9 a 10 áudios.

As referências que irei colocar aqui abaixo possuem dois exemplos de fine-tuning muito interessantes que implementam o pipeline citado anteriormente.

fofr/musicgen-sonic – Run with an API on Replicate: Esse fine-tuning foi feito utilizando apenas a trilha sonora de Sonic 2, o que foi suficiente para se obter resultados bem interessantes.

https://replicate.com/p/36tn2vtby3raqlb7uvnqlrii5i: Esse fine-tuning juntou musicas de coral com músicas de jogos 16bit/8bit. O resultado, também, é muito interessante.

https://replicate.com/blog/fine-tune-musicgen: Referência geral sobre os fine-tunings que foram mostrados.

Com base nesse pipeline e na ferramenta de auto-anotação, a ideia de montar um próprio dataset tornou-se completamente possível. Portanto, para começar, irei montar um pequeno dataset que será composto por um jogo de gênero específico que servirá de base para um primeiro experimento com pipeline da Replicate.

3. **Encontrar o modelo generativo text-to-audio ideal para o TCC**

Para essa etapa, eu levei em conta quais modelos possuem repositórios abertos (não necessariamente oficiais) para o uso e que poderiam ser utilizados para um fine-tuning, visto que o treinamento de um modelo desse necessita de uma grande quantidade de recursos computacionais. Nesse sentido, os modelos que atenderam esses requisitos foram o MusicGen (META) e o MusicLM.

Tabela resumindo alguns pontos dos modelos:

Característica	**MusicGen**	**MusicLM**
Lançamento	Agosto de 2023	Janeiro de 2023
Modelagem	Modelo de linguagem de estágio único	Modelo hierárquico seq2seq
Avaliação de Desempenho	Acima dos baselines atuais	Acima dos baselines da época

Taxa de Amostragem	32 kHz	24 kHz
Áudio Gerado	Mono e estéreo	Estéreo
Entrada	Texto e melodia	Texto e melodia
Recursos Disponíveis	Códigos, modelos e amostras disponíveis publicamente	Dataset MusicCaps com 5.5k pares música-texto disponível e demonstrações
Qualidade de Áudio	Alta qualidade, com demonstrações disponíveis	Alta qualidade, com demonstrações disponíveis
Treinamento	20 mil horas de músicas licenciadas	280 mil horas de treinamento
Acessibilidade	Implementação oficial disponível para treinamento e fine-tuning	Sem implementação oficial disponível, apenas não-oficial
RAM/VRAM	RAM: 20GB (mais ou menos) VRAM: 15GB (mínimo)	Não especificado

Basicamente, o MusicGen e o MusicLM são dois modelos que possuem resultados extremamente interessantes, porém, o fato do MusicGen ser open source e possuir ferramentas que facilitam o seu fine-tuning fez completa diferença no momento de escolher entre ele e o MusicLM. Nesse sentido, a partir de agora o modelo que será profundamente estudado e que servirá de base para os primeiros experimentos será o MusicGen.

Abaixo, estão alguns links que possuem exemplos de áudios gerados pelos dois modelos. Com isso, é possível perceber que os dois estão muito próximos em termos de qualidade.
MusicLM: MusicLM: Generating Music From Text
MusicGen:MusicGen - a Hugging Face Space by facebook (demo para experimentar

o MusicGen)

Links úteis relacionados ao modelo:
MusicGen:

- Repositório que facilita o treinamento e fine-tuning do MusicGen: https://github.com/Stability-AI/stable-audio-tools/tree/main/stable_audio_tools

- Pipeline de fine-tuning com auto-anotação: Fine-tune MusicGen to generate music in any style – Replicate:

- Repositório oficial do MusicGen: MUSICGEN.md - facebookresearch/audiocraft · GitHub

- Notebook feito por mim implementando o pipeline de fine-tuning da stable audio tools: FineTuningMusicGen.ipynb

MusicLM:

- Implementação não oficial do modelo: Implementation of MusicLM, Google's new SOTA model for music generation using attention networks, in Pytorch

Outros modelos avaliados:
Alguns outros grandes modelos foram avaliados, a exemplo do AudioLDM2, Mousai e Riffusion, mas, mesmo com repositórios, não havia acesso a um pipeline de treino ou fine-tuning. Portanto, esses modelos foram descartados temporariamente.

4. Buscar maneiras de condicionar a geração musical à emoção

Como o modelo escolhido foi o MusicGen, o condicionamento do modelo pode ser bem simples, pois existem estruturas na arquitetura que são chamadas de condicionadores. Nesse sentido, são essas as estruturas responsáveis por garantir um controle robusto sobre a saída do modelo e condicionar a geração musical a partir de textos e melodias, logo, pensando nesse cenário, existem duas maneiras de usar a emoção como um fator condicionante.

1. A primeira opção seria a mais simples de se pensar que é utilizar um prompt vinculado a cada música durante o fine-tuning que seja feito baseado na emoção. Por exemplo, a descrição “Uma música vibrante e alegre com ritmos rápidos” serviria de base para condicionar o modelo a gerar uma saída que traga uma emoção vinculada à felicidade. Desse modo, a descrição utilizada de prompt é um fator muito importante no MusicGen, pois o áudio gerado possui uma alta fidelidade com o que foi passado de input.

2. A segunda opção surge do uso do repositório da stable audio tools, pois com ele é possível utilizar outras maneiras de condicionar o modelo durante o fine-tuning. Portanto, outra alternativa é passar a emoção diretamente para o modelo a partir da discretização das emoções referentes a cada música, pois a ferramenta implementa um condicionador que pode receber classes como input e utilizá-las como condicionantes para a geração musical. Com isso, a descrição utilizada no prompt pode ser limitada às características em si da música, enquanto a emoção já estará servindo de condicionante para a saída do modelo.

Considerações: A segunda opção já está parcialmente implementada, pois o pipeline utilizando a stable audio tools está pronto para uso, mas, por falta de recursos computacionais do google colab, não foi possível experimentar a efetividade da abordagem. Além disso, a primeira opção será testada com o pipeline da Replicate, pois a adição de descrições musicais é um pouco mais simples nesse pipeline.

APÊNDICE 3

Termo de Aceite de Entrega 4

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 16 de nov. de 2023

Participantes da Entrega [matriculados em Residência em IA]:

Luiz Fernando de Araújo Vidal

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Entregas previstas para o atual GATE:

- **Buscar novidades apresentadas no ISMIR 2023**
- **Preparar o ambiente para o pipeline de fine-tuning do Replicate e iniciar experimentos**
- **Realizar mais testes com o pipeline de fine-tuning da stable audio tools (Stability-AI)**
- **Montar um pequeno dataset com músicas de um jogo de gênero específico para validar no pipeline do Replicate**

Documento com as entregas: Entrega Gate 16/11/23

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

- **Investigar o problema de API da Replicate**
- **Investigar erros do pipeline da stable audio tools e realizar mais experimentos**
- **Estudar o paper Data Collection in Music Generation Training Sets: A Critical Analysis**
- **Incluir novas emoções no dataset que está sendo montado**

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

LUANA GUEDES BARROS MARTINS: Go!

GATE 16/11/13

Buscar novidades apresentadas no ISMIR 2023
Neste ano, o ISMIR 2023 não trouxe muitos papers na área de generativos, principalmente, na parte de geração de músicas “reais” (no formato de áudio), mas alguns artigos aplicados a games surgiram.
Papers aplicados a games:
LP-5: AutoOsu: Audio-Aware Action Generation for Rhythm Games:
https://ismir2023program.ismir.net/lbd_319.html
LP-22: Beat-Aligned Spectrogram-to-Sequence Generation of Rhythm-Game Charts
https://ismir2023program.ismir.net/lbd_337.html

Outro artigo interessante que apareceu foi o seguinte:
P1-03: Data Collection in Music Generation Training Sets: A Critical Analysis
https://ismir2023program.ismir.net/poster_15.html

Esse dataset fala um pouco sobre a coleta de dados para o treinamento de modelos generativos a partir de uma análise sistemática dos datasets utilizados nas últimas 10 edições do ISMIR. Uma informação interessante que foi levantada no paper é o fato de que 42,6% dos dados utilizados foram extraídos da internet sem nenhum tipo de preocupação com licenças de uso.

Music ControlNet
Como no ISMIR não identifiquei nada muito interessante para o meu projeto, fui atrás de novidades fora da conferência. Nesse sentido, nessa última semana foi lançado o Music ControlNet que é um modelo generativo musical que traz uma proposta muito interessante de controle,”pixel a pixel”, sobre a saída musical utilizando atributos musicais como o posicionamento das batidas e o controle do ritmo.Vale destacar que o modelo supera o MusicGen em benchmarks, sendo 49% mais fiel ao gerar músicas a partir de melodias e, tudo isso, possuindo muito menos parâmetros e dados de treinamento.

Preparar o ambiente para o pipeline de fine-tuning do Replicate e iniciar experimentos

Essa etapa não foi possível de realizar, pois a API não está reconhecendo a minha API key, o que está sendo um problema para outras pessoas também, pois busquei informações no Discord e encontrei outras reclamações sobre isso.

Para usar a API key é necessário colocá-la como variável de ambiente, mas, mesmo fazendo esse processo de diversas maneiras, a API não reconhece. Além disso, recebi cobrança da API apesar de não ter conseguido rodar nada nela, o que vai ser mais um problema para resolver.

Erros:

```
{"title":"Unauthenticated","detail":"You did not pass an authentication token","status":401}
```

Realizar mais testes com o pipeline de fine-tuning da stable audio tools (Stability-AI):

Para esses testes, foi utilizado o dataset inteiro de games 8bits (ISMIR 2022: YM2413-MDB: A Multi-Instrumental FM Video Game Music Dataset with Emotion Annotations). Além disso, utilizei o condicionamento via prompt e emoção, ou seja, o modelo além de receber um prompt baseado na emoção, ele também recebe a emoção discretizada como condicionante.

O objetivo principal desses testes era conseguir rodar um fine-tuning pelo máximo de épocas possíveis sem estourar os limites dos recursos do Google Colab. Logo, parte dessa etapa foi tentar buscar algum problema na chamada da função de treinamento, como bibliotecas que não foram instaladas corretamente ou algum erro de alocação. A partir disso, foi possível descobrir que a biblioteca xtransformers não estava sendo utilizada para otimizar o pipeline de treinamento, pois havia incompatibilidade entre as bibliotecas do colab e do repositório da stable audio tools.

Após instalar, corretamente, a biblioteca, foi possível fazer o fine-tuning por algumas épocas utilizando uma amostra de 50 áudios, mas o problema de falta de recursos veio à tona novamente depois de 5 épocas. Portanto, resolvi trabalhar nos áudios, principalmente, na taxa de amostragem e no tamanho do input do modelo.

Teste 1: 44100hz, 30s de input e 50 audios -> Faltou recursos, logo, o ideal seria reduzir o tamanho da amostra de 30s para 10s e aumentar a quantidade de áudios para ver se é possível treinar por mais épocas.

Teste 2: 44100hz ,10s de input e 357 audios -> 26 épocas (erro de worker killed). O que foi interessante, pois o pipeline gera algumas demos durante o processo.

Teste 3: 32000hz ,10s de input e 669 audios -> 16 épocas (erro de worker killed)

Abaixo estão algumas demos que, apesar de não estarem com áudios de ótima qualidade, já é possível perceber a presença do estilo de música 8bit dos games. O prompt utilizado para gerar as demos foi “cheerful video game music”.

Demo gerada no Teste 2: demo_teste2.wav
Demo gerada até então do Teste 3: demo_teste3.wav

Notebook do Fine-tuning: FineTuningMusicGen.ipynb
Pasta com o dataset: 8bit games

Montar um pequeno dataset com músicas de um jogo de gênero específico para validar no pipeline do Replicate

O dataset foi montado a partir de 3 emoções (Feliz, Triste e Calmo) e de jogos de gêneros variados, pois foi mais simples de conseguir os dados dessa maneira. Logo, eu recorri às playlists de músicas que são possíveis de encontrar no youtube e que, geralmente, possuem descrições referentes ao “humor” da música, como por exemplo, a playlist de música de games tristes que possui o título “The most touching Game Soundtracks - emotional/beautiful/sad OST MIX”. Com isso, foi possível obter algumas horas de músicas já anotadas.

Feliz: 25 musicas, totalizando 58 minutos de áudio

Triste: 19 músicas, totalizando 1 hora e 6 minutos de áudio

Calmo: 30 musicas totalizando 1 hora e 2 minutos de música

Total de horas do dataset: Cerca de 3 horas

Feliz:upbeat | a video game music mix
Triste: The Most Touching Game Soundtracks - emotional / beautiful / sad OST MIX
Calmo: https: "Relaxing" Resident Evil Music

Com isso, seguindo as exigências do pipeline da Replicate, eu gerei um arquivo txt pra cada música com uma descrição baseada na emoção.

- Músicas com a label “Feliz”, possuem a descrição “happy,upbeat,video game music”.
- Músicas com a label “Triste”, possuem a descrição “emotional,touching,sad,video game music”.
- Músicas com a label “Calmo”, possuem a descrição “calm,relaxing,video game music”.

Por fim, coloquei todos os arquivos em uma pasta e os zipei para usar de entrada no pipeline.

Link para do dataset: dataset_v1

Termo de Aceite de Entrega 5

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 23 de out. de 2023

Participantes da Entrega [matriculados em Residência em IA]:

Luiz Fernando de Araújo Vidal

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Entregas previstas para o atual GATE:
- **Investigar o problema de API da Replicate**
- **Investigar erros do pipeline da stable audio tools e realizar mais experimentos**
- **Estudar o paper Data Collection in Music Generation Training Sets: A Critical Analysis**
- **Incluir novas emoções no dataset que está sendo montado**

Link para a entrega: Entrega Gate 23/11/23

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

- Alterar o formato dos áudios do dataset que está sendo construído e adicionar mais músicas de cada emoção
- Realizar o fine tuning com o pipeline da Replicate no dataset de games 8 bit
- Analisar os resultados obtidos pelo primeiro fine tuning
- Realizar o fine tuning com os áudios no formato novo (WAV).

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

LUANA GUEDES BARROS MARTINS: Go!

GATE 23/11/13

1. Investigação do problema de API da Replicate

- Erro que estava impedindo o avanço do fine tuning:

{"title":"Unauthenticated","detail":"You did not pass an authentication token","status":401}

- O problema da API foi resolvido após perceber que o erro apresentado, na verdade, era puramente visual, pois, ao monitorar o fine tuning pelo site da própria API, percebi que os erros apresentados não estavam relacionados com o acesso a API, mas sim, à memória. Nesse sentido, experimentei alterar as GPUs disponíveis pela API e a única que funcionou sem apresentar erro foi a opção mais cara e potente que é um cluster de 8x A40 com 600GB de memória RAM. Com isso, em 14 minutos consegui realizar o fine tuning do MusicGen Medium com o dataset criado por mim.

  Exemplos de resultados obtidos com o dataset criado:
  - calm_relaxing.wav
  - emotional_touching_sad.wav
  - happy.wav
  - tense_horror.wav (relacionada à nova emoção "fear"/"medo" inserida no dataset.)

  Pontos que podem ser melhorados:
  - O formato dos áudios que foram utilizados no fine tuning é o MP3, o que, geralmente, não é o ideal, pois muitas informações de frequências não audíveis são perdidas quando o áudio bruto é convertido para MP3. Nesse sentido, irei refazer o dataset com os áudios no formato WAV para que eu possa obter o máximo de informações de cada áudio.
  - Melhorar a descrição (prompt) que é passada para cada áudio
  - Treinar por mais epochs, pois o fine tuning atual foi feito com 3 epochs
  - Aumentar o dataset

  Link para experimentar o meu fine tuning: game-emotion-musicgen

  Recomendações de prompts:
  - happy,video game music
  - emotional,touching,sad,video game music
  - calm,relaxing,video game music
  - tense,horror,frightening,video game music

2. Investigação dos erros do pipeline da stable audio tools e a realização de mais experimentos

- O erro anterior estava diretamente relacionado à insuficiência de memória RAM para a alocação dos workers. Em busca de soluções, consultei informações no repositório da Stable Audios Tools e notei a possibilidade de alterar a precisão dos

pontos flutuantes usados durante o treinamento. Embora a precisão padrão fosse teoricamente a mais baixa (float16), explorei outras opções na documentação do TensorFlow, que faz parte do backend do repositório, e descobri a opção de 'mixed precision'. Esta técnica é projetada principalmente para otimizar o uso de memória. Com essa mudança, o consumo de memória, que antes era de 11gb, caiu para 9, mas com o tempo a memória continua a encher e o os workers são derrubados.

Contudo, a redução da taxa de amostragem (de 44100 para 32000) e do tamanho das amostras (de 30 segundos para 10 segundos) resultou em uma queda na qualidade dos resultados parciais obtidos durante o fine tuning. Essa situação me leva a concluir que é inviável treinar o modelo MusicGen de forma gratuita no Colab. Essas modificações, especialmente a diminuição do tamanho das amostras, impactam negativamente a qualidade do treinamento. Vale ressaltar que a recomendação para o modelo é o uso de amostras de no mínimo 30 segundos, o que não é possível no cenário atual.

Exemplos do finetuning com 14 epochs (máximo que deu antes dos workers pararem) e mixed precision:

- demo_mixed_precision1.wav
- demo_mixed_precision2.wav

3. **Estudo do paper Data Collection in Music Generation Training Sets: A Critical Analysis**

Para essa entrega, fiz uma breve análise dos principais pontos discutidos no paper.
Análise do paper:

Quantidade de Datasets Analisados:

- Os autores examinaram um total de 315 papers das últimas dez edições do ISMIR (2013-2022), focando nos datasets utilizados para treinar modelos de AMG (Automatic Music Generation).

Métodos de Coleta de Dados:

- Foi identificado que 42,6% dos datasets foram coletados de maneira indiscriminada, sem procurar o consentimento dos músicos ou direitos autorais.
- Apenas uma pequena fração dos datasets mencionou qualquer forma de consentimento obtido dos artistas ou remuneração por suas contribuições.

Tipos de Datasets Utilizados:

- Os dados coletados variam em tipo e tamanho, abrangendo desde pequenas coleções de músicas específicas até grandes conjuntos de dados com milhares de peças musicais.

Exemplos Específicos e Casos de Estudo

- Casos Problemáticos: O artigo destaca exemplos onde o consentimento dos artistas não foi obtido, ilustrando como os dados foram usados de forma potencialmente antiética.

Discussão sobre Práticas de Coleta

- Reflexão Ética na Documentação dos Datasets: Apenas uma minoria dos datasets discutiu questões éticas relacionadas à sua coleta e uso, demonstrando uma falta geral de conscientização e preocupação com a ética na comunidade de AMG.

Recomendações para Melhores Práticas:

- **A pesquisa oferece recomendações específicas para melhorar as práticas éticas, incluindo a criação de novos datasets com consentimento explícito, remuneração justa para os artistas, e maior transparência no processo de coleta de dados.**

Conclusões dos Autores:

- Impacto da Falta de Ética na Coleta de Dados: A pesquisa conclui que a abordagem atual na coleta de dados para AMG pode estar fomentando práticas exploratórias e desconsiderando os direitos dos artistas, ressaltando a necessidade de uma mudança significativa na abordagem do campo.

4. Inclusão de novas emoções no dataset que está sendo montado

Novas músicas relacionadas a emoção “medo” foram adicionadas ao dataset. O processo de extração foi, basicamente, o mesmo das outras emoções, pois utilizei uma playlist de músicas de jogos de terror para extrair aquelas que possuíam uma capacidade maior de gerar um clima tenso e sombrio.

Resultado: 13 músicas extraídas totalizando 1 hora de duração

Dataset final: 77 músicas englobando as emoções happy, sad, fear, calm.

Obs: Vale ressaltar que utilizei emoções derivadas das emoções principais para enriquecer a descrição das músicas.

Dataset com as novas emoções: dataset_v2.zip

Fonte:Horror Game OST's

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

APÊNDICE 4

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

Termo de Aceite de Entrega 6

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 30 de nov. de 2023

Participantes da Entrega [matriculados em Residência em IA]:

Luiz Fernando de Araújo Vidal

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Entregas previstas para o atual GATE:

- **Alterar o formato dos áudios do dataset que está sendo construído e adicionar mais músicas de cada emoção**
- **Realizar o fine tuning com o pipeline da Replicate no dataset de games 8 bit**
- **Analisar os resultados obtidos pelo primeiro fine tuning**
- **Realizar o fine tuning com os áudios no formato novo (WAV).**

Documento gerado: Entrega Gate 30/11/23

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Próximos passos:

- **Estudar melhorias no prompt**
- **Realizar o fine-tuning do MusicGen medium nos dois datasets juntos**
- **Estudar a construção de um minigame para introduzir músicas geradas pelo modelo**

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

Agradecimentos para Luiz Guilherme, Heloisy e Alex que se disponibilizaram para anotar os áudios gerados por mim.

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

LUANA GUEDES BARROS MARTINS: Em análise!

GATE 30/11/13

1. **Alterar o formato dos áudios do dataset que está sendo construído e adicionar mais músicas de cada emoção**
   Como o antigo dataset estava no formato mp3 que, por conta de sua compressão, causa perdas de informações a nível espectral, realizei uma reformulação do dataset para que todos os áudios estivessem no formato WAV. Além disso, foi aproveitado o momento para que mais músicas fossem adicionadas em algumas emoções que, com base em opiniões externas, não estavam sendo 100% representadas na geração musical.

   - **Calm**: 30 para 70 músicas, 2 horas e 23 min (adicionado, basicamente, mais de uma hora)
   - Happy: 25 músicas, (inalterado), 1 hora
   - **Sad**: 19 para 38 músicas, 2 horas (adicionado mais uma hora de música)
   - Fear: 13 Músicas (inalterado), 1 hora
   - Total de horas: Cerca de 6 horas e 23min

   Fontes:
   Calm: "Relaxing" Resident Evil Music (todas as músicas)
   Happy: upbeat | a video game music mix
   Sad: The Most Touching Game Soundtracks - emotional / beautiful / sad OST MIX e The Most Touching Game Soundtracks 2 - emotional / beautiful / sad OST MIX

   Dataset:dataset_v3

2. **Realizar o fine tuning com o pipeline da Replicate no dataset de games 8 bit:**
   Para realizar o fine-tuning do modelo no dataset de músicas de games 8 bit, eu utilizei as emoções anotadas no csv do dataset para gerar as descrições. Nesse sentido, me utilizando da coluna “verifieds_tags”, foi possível construir as descrições que seriam passadas como prompt para condicionar a geração da música durante o fine-tuning.

	fname	verified_tags	toptag_eng_verified	4Q
0	01 - Game de check! Koutsuu Anzen (FM) - Instr...	cheerful, comic	cheerful	Q1
1	01 A Ball of Light.wav	dreamy, bizarre	dreamy	Q4
2	01 Compile.wav	peaceful	peaceful	Q4
3	01 Hyper Defending Force (Title).wav	tense, serious, fluttered, rhythmic	tense	Q2
4	01 Is it Domingo Today.wav	cheerful, speedy, fluttered	cheerful	Q1

Basicamente, gerei os prompts no seguinte formato:

- f'{df["verified_tags"]}, 8bit video game music'
- Exemplo de prompt: cheerful, comic, 8bit video game music

Infelizmente o fine-tuning não saiu como esperado, pois utilizei o modelo stereo-medium (lançado bem recentemente) para buscar resultados melhores a nível de qualidade sonora. Entretanto, aparentemente, a geração musical não está saindo como deveria, pois os áudios gerados não estão saindo com uma boa qualidade rítmica e sonora.
Exemplos da baixa qualidade:
8bit_happy.wav
8bit_calm.wav

Link para experimentar o modelo: 8bit-musicgen-games

3. Analisar os resultados obtidos pelo primeiro fine tuning

Para a análise dos resultados, busquei focar na corretude da emoção, ou seja, se o áudio está representando bem a emoção desejada e se a qualidade do áudio está boa.
Para isso, usando a ferramenta de anotação feita por mim:
Ferramenta_anotação_musicgen.ipynb

5 áudios de cada emoção (Happy, sad, fear e calm) foram anotados por 3 pessoas (Luiz Fernando (eu), Luiz Guilherme e Heloisy).

Valores para as emoções: "Certo" ou "Errado"
Valores para a qualidade dos áudios: "Bom","Medio" e "Ruim"

Análise por emoção:

	base_emotion	emotion_correct_heloisy	emotion_correct_luizf	emotion_correct_luizg
0	calm	100.0	100.0	80.0
1	fear	100.0	100.0	100.0
2	happy	60.0	100.0	100.0
3	sad	40.0	100.0	40.0

A análise do aproveitamento geral para cada emoção base (considerando todos os 5 áudios de cada emoção como um todo) mostra o seguinte:

Calm: Os 3 anotadores apresentaram uma boa concordância com os áudios, o que mostra bons resultados para áudios gerados para essa emoção.

Fear: Todos os anotadores tiveram uma taxa de 100% de concordância, o que demonstra ótimo desempenho do modelo em gerar músicas que despertam o medo.

Happy: Também apresenta um bom desempenho a nível de concordância, pois dois anotadores obtiveram resultados iguais enquanto o outro apenas considerou que 3 áudios tiveram a emoção bem representada. Nesse sentido, pode-se dizer que o modelo está funcionando relativamente bem para áudios que condicionam a emoção de felicidade.

Sad: Percebe-se que para a emoção Sad, o modelo não consegue gerar músicas que representam 100% a tristeza, pois ele obteve um desempenho de 40% com os anotadores Luiz Guilherme e Heloisy. Nesse sentido, isso pode ser explicado pela maneira que a descrição (ou prompt) foi passada para o modelo durante o fine-tuning que não teve foco na tristeza, logo, o prompt “emotional, touching, sad, video game music” pode não ser a melhor maneira de condicionar o modelo a gerar músicas tristes.

Análise por emoção com foco na qualidade:

	base_emotion	quality_good_heloisy	quality_good_luizf	quality_good_luizg
0	calm	100.0	40.0	80.0
1	fear	60.0	80.0	80.0
2	happy	100.0	40.0	60.0
3	sad	100.0	100.0	40.0

Obs: Não ocorreram avaliações de valor “Ruim”, apenas “Bom” e “Medio”

Calm:
Heloisy: 100% dos áudios avaliados como de boa qualidade.
LuizF: 40% dos áudios avaliados como de boa qualidade.
LuizG: 80% dos áudios avaliados como de boa qualidade.

Fear:
Heloisy: 60% dos áudios avaliados como de boa qualidade.
LuizF e LuizG: 80% dos áudios avaliados como de boa qualidade.

Happy:
Heloisy: 100% dos áudios avaliados como de boa qualidade.
LuizF: 40% dos áudios avaliados como de boa qualidade.
LuizG: 60% dos áudios avaliados como de boa qualidade.

Sad:
Heloisy e LuizF: 100% dos áudios avaliados como de boa qualidade.
LuizG: 40% dos áudios avaliados como de boa qualidade.

Esses resultados mostram uma variação considerável na percepção da qualidade

dos áudios entre os anotadores para as diferentes emoções. Como é algo muito subjetivo, a percepção de qualidade pode variar bastante, mas pode-se afirmar que a qualidade dos áudios gerados pelo modelo MusicGen medium é de média para boa.

Link para a demo do modelo:Game-emotion-musicgen
Link para a pasta com os áudios da análise: audios_analise_datasetv2
Link para o notebook da análise: Analise_audios_musicgen.ipynb

4. **Realizar o fine tuning com os áudios no formato novo (WAV).**
Uma observação é que utilizei o musicgen-stereo-medium (o mesmo do fine-tuning em música 8bit) e os resultados também não foram satisfatórios. Talvez, o problema possa ser em alguma etapa de pré-processamento dos áudios que precisa ser feita especificamente para esse tipo de modelo stereo. Por fim, irei retornar para o modelo medium padrão, pois os resultados apresentados por ele foram interessantes.

Exemplos da baixa qualidade:
tense_horror_fear.wav
happy_musicgenstereo.wav
Link para a demo do modelo:
vidalfer/game-emotion-musicgen – Run with an API on Replicate

APÊNDICE 5

Termo de Aceite de Entrega 7

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 7 de dez. de 2023

Participantes da Entrega [matriculados em Residência em IA]:

Luiz Fernando de Araújo Vidal

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Entregas previstas para o atual gate:

- **Estudar melhorias no prompt**
- **Realizar o fine-tuning do MusicGen medium nos dois datasets juntos**
- **Estudar a construção de um minigame para introduzir músicas geradas pelo modelo**

Documento gerado: Entrega Gate 07/12/23

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Próximos passos:

- **Melhorar a arte e o código do minigame**
- **Avaliar os resultados do fine-tuning realizado nos dois datasets juntos**
- **Estudar a playlist Generative Music AI Course - YouTube**

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

LUANA GUEDES BARROS MARTINS: Em análise!

GATE 07/12/13

1. **Estudar melhorias no prompt:**
   Basicamente, no MusicGen, adicionar informações sobre a emoção da música no prompt é uma boa prática para gerar bons resultados, mas também é possível potencializar esses resultados ao adicionar informações sobre tom, tempo, instrumentos e entre outras características musicais. Portanto, irei aproveitar a funcionalidade de auto labeling que o pipeline da Replicate utiliza para incrementar os prompts que eu estou utilizando. Com isso, cada música irá possuir labels de tempo (bpm), tom, gênero, instrumentos e emoção que serão adicionadas durante o fine-tuning para complementar as descrições (prompts) passadas para cada música. Por fim, o resultado gerado é um json com os campos correspondentes às labels e um campo para a descrição que foi feita por mim.

   Referências:
   Unleash Your Musical Creativity: Writing Prompts for MusicGen | by Stefan Silver | MLearning.ai | Medium
   AudioCraft: A Guide to text-to-music AI - Predictea Digital Care - AI Strategy, Predictions, Data Analysis
   GitHub - sakemin/cog-musicgen-fine-tuner

2. **Realizar o fine-tuning do MusicGen medium nos dois datasets juntos**
   Na última entrega foi feito o fine-tuning utilizando o modelo stereo-medium do MusicGen, mas, como o pipeline não estava atualizado, os resultados foram bem ruins. Com isso, alguns dias depois foi lançada uma atualização no pipeline que permite realizar o fine-tuning corretamente no modelo stereo e, portanto, irei experimentar utilizar essa versão do modelo novamente.

   Os resultados, agora, foram bem melhores tanto na qualidade sonora como na representação da emoção.

   Exemplos:
   sad_emotional_melancholy_game_music.wav
   happy_cheerful_152bpm.wav
   dark_tense_game_music.wav
   calm_relaxing_game_music.wav

   Dataset utilizado:
   Músicas de games mais atuais (dataset montado por mim): 146 músicas
   Músicas de games 8bits: 472 músicas
   Total: 618 músicas

   Link para a demo do modelo: Game-Music-Generator

3. **Estudar a construção de um minigame para introduzir músicas geradas pelo modelo:**
   Como não possuo nenhum tipo de experiência com desenvolvimento de games, foi

necessário buscar alguma maneira de contornar esse problema, pois a criação de um jogo é um processo muito trabalhoso tecnicamente e artisticamente. Nesse sentido, com algumas pesquisas, foi possível perceber que o Chat GPT pode ser um grande aliado nesse meu problema, pois com ele eu posso criar o código do meu jogo e também as artes necessárias para os assets do jogo.

Com isso, com base em algumas fontes, consegui iniciar o desenvolvimento de um side scrolling game onde o conceito do jogo é, basicamente, se movimentar pelo cenário e matar o máximo de monstros possível sem sofrer danos, caso contrário, o jogador recebe uma mensagem de “Game Over”. Além disso, a arte do minigame será completamente baseada na série de jogos Castlevania, mais especificamente o Castlevania Symphony of The Night, que possui uma temática de horror gótico 2D.

Prompt para o código base:
“You are a professional video game programmer and an expert in coding side-scroller type video games. Write p5.js code for a parallax side-scroller video game where you move the main character with the left and right arrow keys and destroy monsters by hitting them with a sword with space key. The up arrow key makes the character jump. The down arrow makes the character crouch. If your character collides with a monster, you lose. Monsters should continue to regenerate until the main character collides with a monster. The main character should always start from the left side of the screen and the monsters on the right side of the screen. The monsters should appear one by one, separated by a second, and move towards the main character. I want to use my own textures for the main character, the monsters and the background. The game should scroll horizontally as the player moves through the game world. The background image should not be distorted and should repeat until the main character loses. There should be a number counter on the top right corner of the screen that displays the number of monsters destroyed by the main character.”

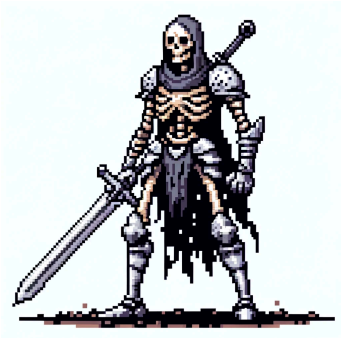
Prompt para as artes:

“Hi I am a game Developer & Designer but art is not my strong part.
I want assets for a 16bit side scroller game on a white background.

The theme is about a vampire killer that hunts vampires and monsters inside a dark castle.“

Artes geradas:

Monstro do jogo:



Caçador de vampiros e monstros:

Cenário (Background):

Minigame base gerado:

Por fim, vale destacar que essas não são as artes finais, pois ainda desejo possuir um acervo mais rico de artes, mas gerar esses assets com o DALL-E é bem complicado por conta da aleatoriedade dele. Além disso, o código gerado é apenas uma base para a implementação de novas funcionalidades dentro do jogo.

Link para o código do jogo feito em Java script:
https://openprocessing.org/sketch/2114081
Referências:
Como fazer prompts para gerar o código:
GPT-4 Mini Game - Ghoul City - Version 1
asteroidsgpt

Como fazer prompts para gerar os assets:
You can generate unlimited game assets in minutes using ChatGPT & DALL-E 3. This is how

Termo de Aceite de Entrega 8

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 14 de dez. de 2023

Participantes da Entrega [matriculados em Residência em IA]:

Luiz Fernando de Araujo Vidal

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Entregas previstas para o atual gate:

- **Melhorar a arte e o código do minigame**
- **Avaliar os resultados do fine-tuning realizado nos dois datasets juntos**
- **Estudar a playlist Generative Music AI Course - YouTube**

Documento gerado: Entrega Gate 14/12/23

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Planejamento para o próximo Gate:

- **Implementar melhorias no minigame**
- **Estudar o pipeline de treino do RAVE (Real Time Audio Variational autoEncoder) para implementá-lo**
- **Buscar por artigos que relacionam a neurociência e geração musical**

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

Queria agradecer Elisa Ayumi, Alex, Heloisy e Luiz Guilherme por contribuírem na avaliação dos resultados do fine-tuning.

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

LUANA GUEDES BARROS MARTINS: Go!

GATE 14/12/13

1. **Melhorar a arte e o código do minigame:**
   Para essa semana, foram implementadas algumas melhorias na arte do jogo e também algumas funcionalidades para o jogo.

   Arte:
   Na arte, a abordagem de utilizar assets gerados pelo GPT, por enquanto, foi deixada de lado por conta da dificuldade de gerar artes que seguem o mesmo padrão. Nesse sentido, eu utilizei assets do próprio Castlevania Symphony of The Night para o meu personagem principal e para os monstros.

   Monstro:

   Caçador/Personagem Principal:

- **Parado:** 
- **Atacando:** 
- **Agachando:** 
- **Pulando:** 
- **Chicote:**

   Com isso, foi possível implementar a animação de pulo, agachamento e ataque. A animação de caminhada ainda está sendo desenvolvida.

   Funcionalidades:

- Barra de vida (HP) no personagem principal e nos monstros
- Sistema de dano, onde o personagem causa 50 de dano por ataque
- Monstros piscam em vermelho ao receber um ataque, pois assim é possível de se confirmar o dano
- Funcionalidade de música de background (Falta apenas resolver um pequeno problema de requisição)

Imagem dentro do jogo:

Código e game: GPT Game
Fonte dos assets: PlayStation - Castlevania: Symphony of the Night - The Spriters Resource

2. **Avaliar os resultados do fine-tuning realizado nos dois datasets juntos:**
Para avaliar alguns resultados obtidos pelo fine-tuning do MusicGen Stereo-Medium, foi utilizada a mesma abordagem da última vez. Nesse sentido, eu gerei 20 músicas para avaliar a geração de música de games atuais e 12 músicas para avaliar a geração de músicas de games 8bits. Vale ressaltar que para o primeiro caso foram 5 áudios de cada emoção (Happy, Fear, Sad e Calm) e no segundo caso foram 3 áudios de cada. Com isso, o foco da avaliação foi, novamente, a corretude das emoções e a qualidade dos áudios gerados.

Ferramenta de anotação: Ferramenta_anotação_musicgen.ipynb
Notebook com as análises: Analise_audios_musicgen.ipynb
Link para a demo do modelo:Music-Game-Generator

Primeiro Caso:

	base_emotion	emotion_correct_heloisy	emotion_correct_luizg	emotion_correct_elisa	emotion_correct_alex	emotion_correct_luizf
0	calm	100.0	100.0	100.0	100.0	100.0
1	fear	100.0	100.0	100.0	100.0	100.0
2	happy	100.0	100.0	100.0	100.0	100.0
3	sad	40.0	80.0	100.0	80.0	100.0

Análise das emoções por anotador:

Calm: Todos os anotadores identificaram corretamente a emoção 'calm' 100% das vezes.

Fear: Houve uma identificação consistente de 'fear', com todos os anotadores acertando 100% das vezes.

Happy: Todos os anotadores também identificaram 'happy' corretamente em todas as ocasiões.

Sad: A emoção 'sad' apresentou variação na identificação. Heloisy teve um desempenho de 40%, e Luiz Guilherme de 80%, enquanto Elisa, Alex e Luiz Fernando mantiveram um desempenho de 100%.

base_emotion	Certo	Errado
calm	100.0	0.0
fear	100.0	0.0
happy	100.0	0.0
sad	80.0	20.0

Porcentagem total de certo e errado por emoção

	base_emotion	quality_good_heloisy	quality_good_luizg	quality_good_elisa	quality_good_alex	quality_good_luizf
0	calm	20.0	100.0	80.0	100.0	100.0
1	fear	20.0	100.0	40.0	100.0	100.0
2	happy	40.0	80.0	100.0	100.0	100.0
3	sad	20.0	80.0	60.0	80.0	100.0

Análise de qualidade por anotador

Calm: Heloisy e Elisa avaliaram 20% e 80% dos áudios 'calm' como 'Bom', respectivamente, enquanto Luiz Guilherme, Alex e Luiz Fernando avaliaram todos (100%) como 'Bom'.

Fear: Heloisy avaliou 20% dos áudios 'fear' como 'Bom', Elisa 40%, e Luiz Guilherme, Alex e Luiz Fernando avaliaram todos (100%) como 'Bom'.

Happy: Heloisy avaliou 40% dos áudios 'happy' como 'Bom', Luiz Guilherme 80%, e Alex, Elisa e Luiz Fernando avaliaram todos (100%) como 'Bom'.

Sad: Heloisy avaliou 20% dos áudios 'sad' como 'Bom', Elisa 60%, Luiz Guilherme 80%, Alex 60% e Luiz Fernando todos (100%) como 'Bom'.

base_emotion	Bom	Medio	Ruim
calm	80.0	20.0	0.0
fear	72.0	20.0	8.0
happy	84.0	16.0	0.0
sad	68.0	24.0	8.0

Porcentagem de cada avaliação de qualidade por emoção

Conclusões Gerais:

- **Consistência nas Emoções:** Os anotadores mostraram uma alta consistência na identificação correta das emoções, exceto para 'sad', o que sugere que essa emoção pode ser mais subjetiva ou difícil de identificar.
- **Variação na Avaliação da Qualidade:** Há uma variação significativa na avaliação da qualidade dos áudios entre os anotadores. Enquanto alguns são consistentemente positivos em suas avaliações, outros são mais críticos. Isso pode indicar uma grande influência da subjetividade nesse tipo de avaliação.

Segundo Caso (Músicas 8bit):

	base_emotion	heloisy_emotion_correct	luizg_emotion_correct	elisa_emotion_correct	alex_emotion_correct	luizf_emotion_correct
0	calm	66.666667	100.0	100.000000	100.0	100.0
1	fear	25.000000	75.0	25.000000	100.0	100.0
2	happy	100.000000	100.0	100.000000	100.0	100.0
3	sad	66.666667	100.0	33.333333	100.0	100.0

Análise das emoções por anotador

Calm: A maioria dos anotadores (Luiz Guilherme, Elisa, Alex, e Luiz Fernando) teve um desempenho de 100% na identificação correta da emoção 'calm', enquanto Heloisy teve um

desempenho de 66.67%.

Fear: A identificação de 'fear' teve uma grande variação. Enquanto Luiz Guilherme, Alex e Luiz Fernando tiveram um desempenho de 75-100%, Heloisy e Elisa tiveram um desempenho de apenas 25%.

Happy: Todos os anotadores tiveram um desempenho de 100% na identificação de 'happy'.

Sad: A identificação de 'sad' variou, com Heloisy e Elisa tendo um desempenho inferior (66.67% e 33.33%, respectivamente) em comparação com os outros anotadores, que tiveram 100%.

base_emotion	Certo	Errado
calm	93.333333	6.666667
fear	68.421053	31.578947
happy	100.000000	0.000000
sad	80.000000	20.000000

Porcentagem total de certo e errado por emoção

	base_emotion	heloisy_quality_good	luizg_quality_good	elisa_quality_good	alex_quality_good	luizf_quality_good
0	calm	66.666667	100.000000	66.666667	100.0	66.666667
1	fear	0.000000	25.000000	0.000000	100.0	25.000000
2	happy	33.333333	100.000000	100.000000	100.0	100.000000
3	sad	0.000000	66.666667	0.000000	100.0	66.666667

Análise das qualidade por anotador

Calm: A qualidade de 'calm' foi avaliada como 'Bom' por Luiz Guilherme e Alex (100%), mas variou entre Heloisy, Elisa e Luiz Fernando (66.66%)

Fear: A qualidade de 'fear' teve uma avaliação mais baixa, com Luiz Guilherme, Alex e Luiz Fernando variando de 25% a 100%, enquanto Heloisy e Elisa não deram nenhuma avaliação 'Bom'.

Happy: A qualidade de 'happy' foi consistentemente avaliada como 'Bom' por todos, exceto Heloisy (33.33%).

Sad: A avaliação da qualidade de 'sad' foi baixa para Heloisy e Elisa, enquanto Luiz Guilherme, Alex e Luiz Fernando variaram de 66.67% a 100%.

base_emotion	Bom	Medio	Ruim
calm	80.000000	13.333333	6.666667
fear	31.578947	5.263158	63.157895
happy	86.666667	13.333333	0.000000
sad	46.666667	40.000000	13.333333

Porcentagem de cada avaliação de qualidade por emoção

Conclusões Gerais para Músicas 8bit:

- **Variação na Identificação de Emoções:** Observa-se uma variação significativa na identificação de algumas emoções, especialmente 'fear' e 'sad', sugerindo uma maior subjetividade ou desafio na identificação dessas emoções específicas nas músicas 8bit.
- **Diferenças na Avaliação da Qualidade:** Há uma variação notável nas avaliações de qualidade entre os anotadores, o que pode indicar uma forte influência da subjetividade nesse tipo de avaliação.

3. Estudar a playlist Generative Music AI Course - YouTube:

Dessa playlist, foi possível extrair um conteúdo interessante na área de geração musical que é o RAVE (Real Time Audio Variational autoEncoder). Com isso, fiz um estudo um pouco mais aprofundado sobre essa arquitetura para entender melhor como eu poderia utilizá-la para gerar áudio a partir de dados não convencionais, como por exemplo, sinais de EEG (eletroencefalograma).

Estrutura Básica:

Autoencoder Variacional (VAE):

O RAVE é um tipo de VAE, que é uma rede neural que aprende a compactar (codificar) dados em um espaço latente de menor dimensão e depois reconstruir (decodificar) esses dados de volta à sua forma original.

Em VAEs, o espaço latente é tratado como uma distribuição probabilística, proporcionando uma maneira de gerar novos dados que seguem a mesma distribuição dos dados de treinamento.

Encoder:
A primeira parte do RAVE é o encoder, que pega um sinal de áudio e o mapeia em um espaço latente.
Este espaço latente representa uma versão comprimida do áudio, onde características importantes são mantidas, mas redundâncias e detalhes desnecessários são descartados.

Decoder:
A segunda parte é o decoder, que reconstrói o áudio a partir das representações latentes. Isso permite gerar áudio que, embora não seja uma cópia exata do original, mantém suas características essenciais.

Características Específicas do RAVE:

Treinamento em Duas Etapas:
O RAVE é primeiro treinado como um VAE regular, focando em aprender uma representação eficiente e significativa do áudio no espaço latente.
Após essa fase, ele passa por um ajuste fino com um objetivo de geração adversária, onde um componente adicional, semelhante às redes generativas adversárias (GANs), é usado para melhorar a qualidade da síntese de áudio.

Multiband Decomposition:
Uma característica notável do RAVE é sua capacidade de decompor sinais de áudio em múltiplas bandas de frequência.
Isso permite que ele opere eficientemente, gerando sinais de áudio de alta resolução (como 48kHz) rapidamente.

Eficiência:
O RAVE pode operar em tempo real ou até 20 vezes mais rápido, dependendo do hardware. Isso o torna adequado para aplicações que requerem geração rápida de áudio, como em performances ao vivo ou em ambientes de produção musical.

Gerando Música com o RAVE:
Pode-se dizer que a geração musical do RAVE é relativamente flexível, especialmente em comparação com sistemas mais tradicionais ou específicos de geração de música. No espaço latente, as características do áudio podem ser manipuladas para alterar ou criar novos sons, isso inclui ajustar elementos como timbre, ritmo e harmonia. Nesse sentido, a manipulação no espaço latente permite a experimentação e a criação de sons únicos que podem ser difíceis de alcançar com métodos tradicionais.

O uso mais direto do RAVE é com dados de áudio padrão, como gravações musicais ou de voz. Aqui, o modelo pode aprender a replicar e variar estilos e gêneros existentes. Por outro lado,conceitualmente, é possível mapear dados de outras modalidades, como sinais EEG (eletroencefalograma) ou informações textuais, para o espaço latente do RAVE. Isso exigiria um processo de conversão desses dados em parâmetros compreensíveis pelo modelo. Por exemplo, as entradas podem ser baseadas em características específicas desejadas, como

um certo humor ou estilo musical, que podem ser mapeadas para ajustes correspondentes no espaço latente. Além disso, para entradas mais complexas ou abstratas, o RAVE pode ser integrado com outros modelos ou sistemas, como modelos especializadas em processamento de texto ou análise de sinais, para converter essas entradas em algo que o RAVE possa processar.

Por fim, o RAVE oferece uma arquitetura flexível e poderosa para a geração e manipulação de áudio e música. Sua capacidade de gerar áudio de alta qualidade, combinada com a habilidade de manipular o espaço latente, torna-o uma ferramenta valiosa para uma ampla gama de aplicações musicais, desde a replicação de estilos existentes até a criação de novas formas de expressão sonora. A utilização de diferentes tipos de entradas para gerar música com o RAVE abre um vasto campo de possibilidades criativas e experimentais, embora muitas dessas aplicações possam exigir um nível de pesquisa um pouco maior, como no caso dos sinais de EEG.

Referência:

20. Audio generation with RAVE - Generative Music AI
[2111.05011] RAVE: A variational autoencoder for fast and high-quality neural audio synthesis
GitHub - acids-ircam/RAVE: Official implementation of the RAVE model: a Realtime Audio Variational autoEncoder

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

APÊNDICE 6

Termo de Aceite de Entrega 9

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 21 de dez. de 2023

Participantes da Entrega [matriculados em Residência em IA]:

Luiz Fernando de Araujo Vidal

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Entregas previstas para o atual GATE:

- **Implementar melhorias no minigame**
  Neste primeiro item, implementei algumas melhorias na arte do jogo ao adicionar cenários que seguem uma progressão de acordo com o score do jogador, ou seja, o jogador começa em um campo aberto, depois vai para a entrada do castelo e para dentro do castelo. Além disso, resolvi o problema de não conseguir inserir música no game.
- **Estudar o pipeline de treino do RAVE (Real Time Audio Variational autoEncoder) para implementá-lo.**
  Neste segundo item, utilizei a implementação oficial do pipeline de treino do RAVE para iniciar alguns experimentos com o modelo. Entretanto, por conta de alguns problemas (estão detalhados no documento gerado) não foi possível realizar o treinamento em tempo hábil, mas o pipeline está, atualmente, 100% funcional.
- **Buscar por artigos que relacionam a neurociência e geração musical.**
  Neste terceiro e último item, busquei por artigos que relacionam, de alguma maneira, a geração musical com a neurociência com o objetivo de entender se há a possibilidade de condicionar a geração de músicas a partir de dados biológicos. Por fim, cada artigo foi brevemente analisado de acordo com alguns pontos que estão destacados no documento gerado.

Documento gerado: Entrega Gate 21/12/23

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Planejamento para o próximo GATE:

- **Revisar o que foi feito no fine-tuning do MusicGen Stereo-medium, com o objetivo**

de identificar quais melhorias podem ser feitas para que o modelo tenha uma performance melhor para músicas 8bits.
- **Realizar experimentos com o RAVE**
- **Refazer o fine-tuning do MusicGen com a versão Melody, pois é possível utilizar a melodia das músicas como condicionante.**
- **Finalizar o desenvolvimento do minigame e inserir uma música gerada por mim**

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

LUANA GUEDES BARROS MARTINS: Go!

GATE 21/12/13

1. **Implementar melhorias no minigame**
   Para essa entrega, foquei em resolver o problema da inserção de música no game, pois, por algum motivo, isso estava causando um problema de requisição e o jogo não executava. Com isso, descobri que o problema não era o código, mas sim um ícone de som que a própria plataforma possui, pois ele indicava que o som estava funcionando, mas na verdade não estava, logo, cliquei duas vezes nele e tudo voltou a funcionar normalmente.

   Além disso, adicionei mais artes ao jogo. Agora, os cenários mudam de acordo com a quantidade de score que o jogador tem e a variação segue uma lógica de progresso, ou seja, o jogador começa em um campo aberto, vai para a entrada do castelo e depois para dentro do castelo.

   Artes de cenário novas:

   Cenário 1:

Cenário 2:

Cenário 3:

A partir de agora, irei focar na adição de novos inimigos, animação de movimentação do personagem e efeitos sonoros.

Link para o game:GPT Game

2. **Estudar o pipeline de treino do RAVE (Real Time Audio Variational autoEncoder) para implementá-lo.**
   O foco desta entrega era deixar completamente funcional o pipeline de treinamento do RAVE, ou seja, resolver possíveis problemas que, com certeza, apareceriam durante a implementação do pipeline e também entender o que é feito em cada etapa do processo de treinamento. Desse modo, a implementação do pipeline de treinamento pode ser encontrada no repositório oficial GitHub - acids-ircam/RAVE: Official implementation of the RAVE model: a Realtime Audio Variational autoEncoder.
   Basicamente, o pipeline é composto por 3 etapas que é o pré-processamento (ou preparação do dataset), treino e exportação do modelo. Nesse sentido, é possível executar o treinamento do modelo no Google Colab, mas, como é um treinamento longo, essa opção torna-se inviável caso não seja utilizada a versão paga. Portanto, o pipeline está completamente funcional, mas precisarei executar localmente por alguns dias para obter bons resultados do treinamento.

   Vale destacar que o pipeline, inicialmente, não estava funcionando, pois a versão atual da biblioteca está apresentando alguns problemas, logo, foi necessário realizar o downgrade da versão 2.3.1 para a versão 2.1.1. Alguns desses problemas foram discutidos em uma “issue” recente que foi criada no repositório oficial do RAVE e, a partir disso, foi possível executar o pipeline.

   Link para a issue: https://github.com/acids-ircam/RAVE/issues/273
   Notebook com o código do treinamento: **Rave training.ipynb**

3. **Buscar por artigos que relacionam a neurociência e geração musical.**
   Como foi dito na apresentação do último gate, essa busca por artigos seria uma maneira de dar os meus primeiros passos para a minha pós graduação. Nesse sentido, eu procurei por artigos que relacionassem geração de músicas com a neurociência, pois eu pretendo entender se há alguma maneira de usar dados biológicos como condicionantes para a geração de músicas.

   Por fim, abaixo serão listados e descritos, brevemente, três artigos que se mostraram interessantes para os meus estudos.

   - **Análise do Artigo "Brain2Music: Reconstructing Music from Human Brain Activity"**

     Descrição do que foi feito
     Este artigo introduz um método para reconstruir música a partir da atividade

cerebral humana, usando imagens por ressonância magnética funcional (fMRI). Eles empregaram o modelo MusicLM para geração de música condicionado em embeddings derivados de dados de fMRI. O objetivo era gerar música que se assemelhasse aos estímulos musicais experimentados pelos humanos em termos de propriedades semânticas, como gênero, instrumentação e humor.

Dados Utilizados
Foram utilizados dados de fMRI coletados de cinco participantes enquanto ouviam clipes de música.

Método Utilizado
O método envolve a previsão de embeddings musicais baseados em dados de fMRI e a subsequente geração ou recuperação de música usando esses embeddings. Dois processos foram explorados: a recuperação de música de um banco de dados existente e a geração de música usando o modelo MusicLM.

Resultados Obtidos
Os resultados indicaram que a música reconstruída era semanticamente semelhante aos estímulos originais, com respeito ao gênero, estilo vocal e humor geral. No entanto, a estrutura temporal do estímulo original muitas vezes não era preservada na reconstrução, e houve casos de falhas com reconstruções de gêneros completamente diferentes.

Conclusões
Os autores concluíram que o método proposto é capaz de extrair informações musicais dos dados de fMRI e gerar música que reflete os estímulos musicais originais a um nível semântico. Além disso, eles investigaram a conexão entre um modelo de geração de música baseado em texto e o cérebro humano, fornecendo uma interpretação quantitativa da representação cerebral de informações musicais semânticas e acústicas.

- **Análise do Artigo "Music Generation Based on Emotional EEG"**

  Descrição do que foi feito
  Este artigo apresenta um método para gerar música baseada em EEG emocional, usando a memória de longo curto prazo (LSTM) para treinar geradores de música emocional. O objetivo é transformar EEGs que refletem diferentes estados emocionais em música, proporcionando uma forma nova de expressar emoções através da música.

  Dados Utilizados

Os dados de EEG emocional utilizados foram retirados do conjunto de dados DEAP, um conjunto público de EEG emocional. Além disso, para treinar o modelo gerador de música, utilizou-se o conjunto de dados de música emocional EMOPIA, que contém clipes de música categorizados em diferentes tipos emocionais.

Método Utilizado
O método envolve três componentes principais: reconhecimento de emoção, estabelecimento de um gerador de música emocional e geração de música baseada em EEG emocional. O sistema utiliza LSTM para capturar a dependência temporal dos dados de música e um modelo SVM para classificar emoções com base no EEG.

Resultados Obtidos
O estudo demonstrou que a música gerada pelo método proposto pode produzir respostas emocionais semelhantes às da música emocional bruta. Além disso, o método mostrou-se competitivo em termos de musicalidade, embora os resultados não tenham sido particularmente satisfatórios em alguns aspectos.

Conclusões
O artigo conclui que o método proposto oferece uma direção interessante para gerar música emocional a partir de EEGs. No futuro, os autores planejam gerar música com acordes e outras formas mais ricas, além de expandir a aplicação de métodos de inteligência artificial para a música baseada em EEG.

Análise do Artigo "Music Generation and Emotion Estimation from EEG Signals for Inducing Affective States"

Descrição do que foi feito
O artigo apresenta um sistema que gera música a partir de sinais de EEG para induzir estados emocionais. A abordagem se baseia na estimação de emoções (valência e excitação) a partir de EEG e na geração de música que corresponde a essas emoções estimadas. O objetivo é induzir emoções personalizadas nos ouvintes.

Dados Utilizados
Os dados utilizados incluíram sinais de EEG de 20 estudantes universitários (10 homens e 10 mulheres). Além disso, avaliações subjetivas das emoções sentidas ao ouvir a música gerada foram coletadas.

Método Utilizado
O sistema consiste em um gerador de música que cria música apropriada para induzir emoções e um estimador de emoção que estima emoções a

partir de sinais de EEG. Nesse sentido, a geração de música foi baseada em cinco parâmetros (tempo, ritmo, volume, tom e modo) calculados a partir dos valores de valência e excitação. Por fim, Três modelos de regressão foram comparados para a estimação de emoções: regressão linear, rede neural convolucional (CNN) e CNN com transfer learning.

Resultados Obtidos

Os resultados mostraram que a música gerada podia efetivamente induzir emoções nos ouvintes.Com isso, a medição da correlação entre as emoções pretendidas pela entrada do gerador de música e as emoções sentidas pelos ouvintes foi razoavelmente alta. Além disso, o modelo CNN com transfer learning obteve os menores erros quadráticos médios (RMSE) entre os valores emocionais reais e estimados, tanto para valência quanto para excitação.

Conclusões

Os autores concluíram que o sistema proposto pode expressar emoções como valores contínuos e gerar música a partir de sinais de EEG de forma eficaz. O estudo também mostrou que a CNN com transfer learning é mais eficaz do que a regressão linear para estimar emoções a partir de EEG.

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

Termo de Aceite de Entrega 10

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 11 de jan. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

Luiz Fernando de Araújo Vidal

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Entregas para o GATE:

- **Revisar o que foi feito no fine-tuning do MusicGen Stereo-medium, com o objetivo de identificar quais melhorias podem ser feitas para que o modelo tenha uma performance melhor para músicas 8bits.**
- **Realizar experimentos com o RAVE**
- **Refazer o fine-tuning do MusicGen com a versão Melody, pois é possível utilizar a melodia das músicas como condicionante.**
- **Finalizar o desenvolvimento do minigame e inserir uma música gerada por mim.**

Documento gerado: Gate 11/01/2024

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Desenvolver o TCC.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

LUANA GUEDES BARROS MARTINS: Em análise!

GATE 11/01/2024

1. **Revisar o que foi feito no fine-tuning do MusicGen Stereo-medium, com o objetivo de identificar quais melhorias podem ser feitas para que o modelo tenha uma performance melhor para músicas 8bits.**

   Para essa tarefa, eu revisei a abordagem que utilizei para montar os prompts das músicas de games 8bit. Como o dataset dessas músicas já é anotado, aproveitei a coluna **“**verifieds_tags” do csv de metadados dos áudios para montar os prompts que alimentariam o modelo durante o fine-tuning.

	fname	verified_tags	toptag_eng_verified	4Q
0	01 - Game de check! Koutsuu Anzen (FM) - Instr...	cheerful, comic	cheerful	Q1
1	01 A Ball of Light.wav	dreamy, bizarre	dreamy	Q4
2	01 Compile.wav	peaceful	peaceful	Q4
3	01 Hyper Defending Force (Title).wav	tense, serious, fluttered, rhythmic	tense	Q2
4	01 Is it Domingo Today.wav	cheerful, speedy, fluttered	cheerful	Q1

Colunas do csv de metadados

Nesse sentido, antes eu apenas formatava uma descrição da seguinte maneira:

- f’{df[“verified_tags”]}, 8bit video game music’

O problema desse método é que músicas de mesma emoção não teriam uma mesma descrição, a exemplo do que foi feito para o outro dataset montado por mim. Com isso, o modelo não teria como aprender o estilo de uma emoção se as músicas que a representam possuem descrições distintas, sendo assim, esse pode ter sido o motivo do baixo desempenho do MusicGen em gerar músicas de games 8bit.

Solução proposta: A solução proposta foi utilizar a coluna toptag_eng_verified para gerar os prompts, mas, mesmo assim, os prompts ficariam muito diversificados para músicas de mesma emoção, logo, realizei o mapeamento das emoções presentes no dataset de games 8bits com as emoções padrões que utilizei no dataset criado por mim (happy, sad, fear e calm).

Mapeamento: O mapeamento foi feito nas emoções abaixo:

```
array(['cheerful', 'tense', 'creepy', 'depressed', 'rhythmic', 'bizarre',
       'peaceful', 'speedy', 'fluttered', 'serious', 'touching', 'grand',
       'dreamy', 'calm', 'boring', 'cute', 'cold'], dtype=object)
```

Emoções do dataset 8bit

Com base nisso, o seguinte mapeamento foi feito:

- Emoção Fear: Depressed, Tense, Serious, Fluttered, Creepy, Bizarre.
- Emoção Calm: Peaceful, Dreamy, Calm, Boring, Rhythmic.
- Emoção Sad: Touching, Cold.
- Emoção Happy: Grand, Cute, Cheerful, Comic, Speedy.

Por fim, padronizei os prompts seguindo a lógica das descrições que utilizei para o dataset que criei.

- Happy: happy, cheerful, 8bit game music
- Sad: sad,emotional,melancholy,8bit game music
- Calm: calm, relaxing, 8bit game music
- Fear: tense, dark, frightening, 8bit game music

Dataset atualizado: dataset_v5.zip

2. **Realizar experimentos com o RAVE**

Essa tarefa não foi possível de se concluir, pois a biblioteca do modelo RAVE está com um problema no momento de exportar modelo treinado. Nesse sentido, realizei o treinamento do RAVE por 50 epochs que levaram cerca de 15 horas e, infelizmente, não consegui utilizá-lo, pois o script de exportação não está funcionando.

Issue com problema idêntico ao meu:
https://github.com/acids-ircam/RAVE/issues/275

Alterei a versão diversas vezes para tentar solucionar o problema, mas mesmo assim não foi possível solucionar. Além disso, para realizar o treinamento precisei utilizar a versão 2.3.0, pois a 2.1.1 que tinha sido a solução dos problemas passados já não estava funcionando mais.

Notebook atualizado: Rave training.ipynb
Checkpoint do modelo 🗀 RAVE

3. **Refazer o fine-tuning do MusicGen com a versão Melody, pois é possível utilizar a melodia das músicas como condicionante.**

Para essa tarefa, o objetivo era realizar o fine-tuning com o MusicGen Stere-Melody que é, basicamente, a versão medium, mas é possível usar áudios como input durante a inferência. Nesse sentido, não há diferença de qualidade ou tamanho entre eles, pois ambos possuem 1.5 bilhões de parâmetros.

Portanto, utilizei o dataset atualizado para verificar se o desempenho para músicas 8 bits iria melhorar com as mudanças nos prompts e pode-se dizer que os resultados foram bem interessantes. A avaliação das músicas foi feita seguindo os mesmo parâmetros de corretude da emoção e qualidade do áudio, mas dessa vez não contará com a avaliação dos anotadores que me ajudaram. Com isso, foram geradas 20 amostras (5 de cada emoção) para que esses áudios fossem anotados.

Correção da Emoção (em %):

emotion_correct	Certo
emotion	
calm	100.0
fear	100.0
happy	100.0
sad	100.0

Tabela de corretude

Calm, Fear, Happy, Sad: 100% dos áudios foram classificados como "Certo" para cada uma dessas emoções.

quality_good	Bom	Medio	Ruim
emotion			
calm	80.0	20.0	0.0
fear	60.0	20.0	20.0
happy	100.0	0.0	0.0
sad	100.0	0.0	0.0

Tabela de Qualidade

Qualidade da Emoção (em %):

- Calm: 80% dos áudios foram classificados como "Bom" e 20% como "Médio".
- Fear: 60% "Bom", 20% "Médio", e 20% "Ruim".
- Happy e Sad: 100% das gravações para ambas as emoções foram classificadas como "Bom".

Portanto, pode-se perceber que houve uma melhora significativa, principalmente, na qualidade dos áudios, pois no fine-tuning anterior foi quase um consenso que a qualidade dos áudios estavam ruins.

Link para os áudios: audios_analise_datasetv5_8bit

Link para o notebook de análise: Analise_audios_musicgen.ipynb
Link para a ferramenta de anotação: Ferramenta_anotação_musicgen.ipynb

4. **Finalizar o desenvolvimento do minigame e inserir uma música gerada por mim.**
   Nesta tarefa, foram implementadas as seguintes melhorias para finalizar o minigame:
   - Implementação de repulsão no personagem quando ele recebe dano
   - Ajuste de valores de dano, pois agora o personagem recebe 20 de dano por ataque
   - Implementação de um chefe que atira bolas de fogo
   - Animação de ataque do chefe
   - Implementação da lógica de disparo das bolas de fogo e da detecção de colisão
   - Implementação da lógica de colisão com o chefe para que o jogador possa causar dano ao chefe
   - Implementação da barra de vida no chefe que nascerá com 500 de vida e perderá 20 de vida por ataque do personagem

   Arte do chefe:

   Arte da bola de fogo:

Agora, o jogo possui um funcionamento bem simples, onde o jogador movimenta o personagem pelas setinhas do teclado e ataca no espaço. A cada 10 pontos de score o cenário muda e, ao atingir 40 pontos, o chefe do jogo irá aparecer. Após derrotá-lo, os monstros do jogo voltam a nascer normalmente.

Por fim, gerei uma música utilizando o modelo que realizei o fine-tuning e utilizei o condicionamento por melodia para gerar uma música no estilo que eu queria. Além disso, utilizei a funcionalidade de gerar uma continuação para a música que obtive de output, logo, consegui um áudio de 1 minuto ao concatenar as duas músicas geradas.

- Música utilizada para condicionar a geração:
  Castlevania - Symphony of the Night - Crystal Teardrops (Original Version)
- Prompt utilizado: tense,dark,frightening, 8bit game music
- Música gerada: game_music_definitive.wav
- Continuação da música: game_music_continuation.wav
- Música final: game_music_final.mp3

Link para o jogo:Castlevania Symphony of GPT

Termo de Aceite de Entrega 11

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 17 de jan. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

Luiz Fernando de Araújo Vidal

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Reavaliação dos últimos resultados obtidos para músicas de games 8 bits

- Como entrega para esse gate extra, eu busquei refazer as avaliações feitas sobre os resultados obtidos para a geração de músicas de games 8 bit após as mudanças nos prompts de treino (Gate 10). Para isso, contei com a ajuda de 3 anotadores para tornar as avaliações mais robustas e coerentes.

Documento gerado: Entrega Gate 17/01/2024

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Desenvolvimento do TCC

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

Gostaria de agradecer o apoio da Elisa Ayumi, Heloisy e Luiz Guilherme , pois se disponibilizaram para anotar os áudios gerados por mim.

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Em análise!

LUANA GUEDES BARROS MARTINS: Go!

GATE 17/01/2024

1. Reavaliação, com a ajuda de anotadores, dos resultados obtidos nas geração de músicas de games 8 bit após as mudanças realizadas no prompt.

Para essa tarefa, contei com a ajuda de Elisa Ayumi, Heloisy Pereira e Luiz Guilherme Correa. Com isso, a avaliação seguiu o mesmo objetivo de verificar se as emoções nas músicas geradas estão corretas e se a qualidade sonora está boa.

Critérios para emoção: Certo e Errado
Critérios para qualidade: Bom, Médio e Ruim.

Link para o modelo: [Game-Music-Generator]
Link para a ferramenta de anotação: Ferramenta_anotação_musicgen.ipynb
Link para o notebook com as análises: Analise_audios_musicgen.ipynb

Avaliações por anotador:

	base_emotion	heloisy_emotion_correct	luizg_emotion_correct	elisa_emotion_correct	luizf_emotion_correct
0	calm	100.0	100.0	100.0	100.0
1	fear	100.0	80.0	100.0	100.0
2	happy	100.0	100.0	100.0	100.0
3	sad	80.0	80.0	100.0	100.0

correção das emoções por anotador

	base_emotion	heloisy_quality_good	luizg_quality_good	elisa_quality_good	luizf_quality_good
0	calm	40.0	100.0	100.0	80.0
1	fear	0.0	100.0	80.0	60.0
2	happy	20.0	80.0	80.0	100.0
3	sad	20.0	100.0	100.0	100.0

Porcentagem de “Bom” por anotador

Os dados mostram que, de forma geral, a correção da emoção foi bastante alta para todas as emoções e anotadores, com exceção de algumas avaliações mais baixas para "Triste" por parte de Heloisy e Luiz Guilherme. Quanto à qualidade, há uma variação maior entre os anotadores. Heloisy tendeu a dar avaliações mais baixas de qualidade, especialmente para as emoções "Medo" e "Feliz". Luiz Guilherme, Elisa e Luiz Fernando deram avaliações mais altas no geral.

Avaliações no geral:

emotion_correct emotion	Certo	Errado
calm	100.0	0.0
fear	95.0	5.0
happy	100.0	0.0
sad	90.0	10.0

Porcentagem de Certo e errado por emoção

quality_good emotion	Bom	Medio	Ruim
calm	80.0	20.0	0.0
fear	60.0	30.0	10.0
happy	70.0	25.0	5.0
sad	80.0	20.0	0.0

Porcentagem de Bom, Médio e Ruim por emoção

Esses resultados indicam que, de forma geral, a correção da emoção foi bastante alta, com exceção de algumas avaliações mais baixas para "Triste" e "Medo". Em termos de qualidade, a emoção "Medo" teve a avaliação mais baixa, com 10% classificado como "Ruim", seguido por "Feliz" com 5% classificado como "Ruim". As emoções "Calma" e "Triste" tiveram avaliações de qualidade mais altas, com a maioria classificada como "Bom".

Comparativo com os resultados gerais anteriores:

base_emotion	Certo	Errado
calm	93.333333	6.666667
fear	68.421053	31.578947
happy	100.000000	0.000000
sad	80.000000	20.000000

Porcentagem total de certo e errado por emoção

base_emotion	Bom	Medio	Ruim
calm	80.000000	13.333333	6.666667
fear	31.578947	5.263158	63.157895
happy	86.666667	13.333333	0.000000
sad	46.666667	40.000000	13.333333

Porcentagem de cada avaliação de qualidade por emoção

Comparação da Qualidade da Música:

Anteriormente:

- Calma: 80% Bom, 13.33% Médio, 6.67% Ruim
- Medo: 31.58% Bom, 5.26% Médio, 63.16% Ruim
- Feliz: 86.67% Bom, 13.33% Médio, 0% Ruim
- Triste: 46.67% Bom, 40% Médio, 13.33% Ruim

Atualmente:

- Calma: 80% Bom, 20% Médio, 0% Ruim
- Medo: 60% Bom, 30% Médio, 10% Ruim
- Feliz: 70% Bom, 25% Médio, 5% Ruim
- Triste: 80% Bom, 20% Médio, 0% Ruim

Análise:

- Calma: A qualidade "Bom" permanece a mesma, mas observamos um aumento na categoria "Médio" e uma diminuição na categoria "Ruim".
- Medo: Houve um aumento significativo na categoria "Bom" e uma redução

nas categorias "Médio" e "Ruim", indicando uma melhora geral na percepção da qualidade.
- Feliz: Observamos uma diminuição na qualidade "Bom" e um aumento nas categorias "Médio" e "Ruim".
- Triste: Aumento na categoria "Bom", diminuição na categoria "Médio" e eliminação da categoria "Ruim".

Comparação da correção da Emoção

Anteriormente:
- Calma: 93.33% Certo, 6.67% Errado
- Medo: 68.42% Certo, 31.58% Errado
- Feliz: 100% Certo, 0% Errado
- Triste: 80% Certo, 20% Errado

Atualmente:
- Calma: 100% Certo, 0% Errado
- Medo: 95% Certo, 5% Errado
- Feliz: 100% Certo, 0% Errado
- Triste: 90% Certo, 10% Errado

Análise:
- Calma: Observamos uma melhoria na correção, passando para 100% "Certo".
- Medo: Melhoria significativa na correção, reduzindo a taxa de "Errado".
- Feliz: Continua com 100% "Certo".
- Triste: Melhoria na correção, com redução na taxa de "Errado".

Conclusão Geral
Em comparação com os resultados anteriores, observamos uma melhoria geral na percepção da correção das emoções, especialmente nas emoções "Calma" e "Medo". Quanto à qualidade, há variações em todas as emoções, com melhorias notáveis em "Medo" e "Triste". Com isso, as mudanças realizadas nos prompts das músicas de games 8 bits realmente trouxeram uma melhora interessante para a geração musical.