

UNIVERSIDADE FEDERAL DE VIÇOSA

Carmengen: um modelo de linguagem para geração de poemas

Maurilio de Araujo Possi
Doctor Scientiae

VIÇOSA - MINAS GERAIS
2024

MAURILIO DE ARAUJO POSSI

Carmengen: um modelo de linguagem para geração de poemas

Tese apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, para obtenção do título de *Doctor Scientiae*.

Orientador: Alcione de Paiva Oliveira

**VIÇOSA - MINAS GERAIS
2024**

**Ficha catalográfica elaborada pela Biblioteca Central da Universidade
Federal de Viçosa - Campus Viçosa**

T

P856c
2024 Possi, Maurilio de Araujo, 1984-
CarmenGen: um modelo de linguagem para geração de
poemas / Maurilio de Araujo Possi. – Viçosa, MG, 2024.
1 tese eletrônica (82 f.): il.

Orientador: Alcione de Paiva Oliveira.
Tese (doutorado) - Universidade Federal de Viçosa,
Departamento de Informática, 2024.

Referências bibliográficas: f. 77-82.

DOI: <https://doi.org/10.47328/ufvbbt.2025.528>

Modo de acesso: World Wide Web.

1. Inteligência artificial. 2. Processamento de linguagem
natural (Computação). 3. Linguagem computacional.
4. Aprendizado do computador. 5. Poesia por computador.
I. Oliveira, Alcione de Paiva, 1962-. II. Universidade Federal de
Viçosa. Departamento de Informática. Programa de
Pós-Graduação em Ciência da Computação. III. Título.

CDD 22. ed. 006.3

MAURILIO DE ARAUJO POSSI

Carmengen: um modelo de linguagem para geração de poemas

Tese apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, para obtenção do título de *Doctor Scientiae*.

APROVADA: 18 de novembro de 2024.

Assentimento:

Maurilio de Araujo Possi
Autor

Alcione de Paiva Oliveira
Orientador

Essa tese foi assinada digitalmente pelo autor em 18/08/2025 às 20:31:15 e pelo orientador em 22/08/2025 às 15:39:53. As assinaturas têm validade legal, conforme o disposto na Medida Provisória 2.200-2/2001 e na Resolução nº 37/2012 do CONARQ. Para conferir a autenticidade, acesse <https://siadoc.ufv.br/validar-documento>. No campo 'Código de registro', informe o código **LFR3.F8O9.L17V** e clique no botão 'Validar documento'.

A minha esposa, Renata Maria.

AGRADECIMENTOS

Primeiramente, agradeço a Deus, o Verbo Eterno, que tudo fez com a Sua Palavra e sustenta todas as coisas com a Sua Vontade. Se esta tese trata da geração de textos, é somente porque o próprio Deus, com sua Palavra, deu origem a todo o universo. A Ele toda honra e glória. Agradeço também à intercessão materna de Nossa Senhora e à proteção de São José, que sempre me acompanharam e guardaram em todos os momentos.

À minha esposa, que sempre esteve ao meu lado incondicionalmente, e não foi diferente nestes cinco anos de doutorado, enfrentando comigo inúmeros desafios, tanto acadêmicos quanto pessoais. Seu amor, paciência, apoio e fé foram inspiração em todos os momentos, e esta conquista é também sua.

Aos meus pais, pelo amor, apoio e ensinamentos que formaram a base do meu caminho. À minha irmã, ao meu cunhado, a minha sogra, tias e demais familiares, pela presença e encorajamento constantes ao longo desta jornada.

Ao meu orientador, Prof. Dr. Alcione de Paiva Oliveira, pela confiança depositada em mim desde os tempos da graduação e pelo constante incentivo a seguir o caminho acadêmico. Se hoje percorro esta trajetória, é porque ele acreditou em meu potencial e me abriu as portas para o mundo científico que estava fora da minha perspectiva. Agradeço também a sua esposa, Dra. Alexandra Moreira, por todo apoio e incentivo.

Aos membros da banca examinadora da defesa desta tese, pela leitura atenta, pelas valiosas contribuições e pelo reconhecimento do mérito deste trabalho. Suas observações e sugestões foram fundamentais para aprimorar a qualidade e a clareza desta pesquisa, enriquecendo não somente este trabalho, mas também minha formação como pesquisador.

Ao Departamento de Informática, pela estrutura, apoio e ambiente acadêmico que tornaram este trabalho possível. Agradeço a todos os professores do departamento pela excelência no ensino, pela inspiração constante e pelo incentivo à pesquisa, bem como aos funcionários que contribuem diariamente para o bom funcionamento das atividades.

Aos colegas que tornaram o caminho mais leve e produtivo, em especial ao Lucas Mucida e ao Cleyton, pela colaboração, troca de ideias e amizade

durante o doutorado.

Aos médicos Dr. Bruno Santos Soares, Dr. Victor Atsushi Kasuya Barbosa e Dr. Flávio Gilbert, pela atenção, dedicação e cuidado comigo e com minha esposa em momentos delicados que enfrentamos nessa jornada.

À Universidade Federal de Viçosa, em especial à Diretoria de Tecnologia da Informação (DTI) na pessoa de sua diretora Licia Felix de Andrade Ramalho, pelo incentivo e apoio para que este trabalho pudesse ser concretizado.

Este trabalho foi realizado com o apoio das seguintes agências de pesquisa brasileiras: Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001, Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG) e Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

RESUMO

POSSI, Maurilio de Araujo, D.Sc., Universidade Federal de Viçosa, novembro de 2024. **Carmengen: um modelo de linguagem para geração de poemas.** Orientador: Alcione de Paiva Oliveira.

A geração automática de textos pela Inteligência Artificial (IA) tem avançado significativamente nas últimas décadas, com modelos capazes de produzir descrições, respostas e narrativas coerentes em diversos contextos. No entanto, a linguagem humana, com sua complexidade estrutural, riqueza semântica e sutileza estilística, permanece como um campo de estudo extremamente desafiador dentro do Processamento de Linguagem Natural. Entre as tarefas mais complexas, a geração automática de poemas ocupa posição singular, sendo reconhecida como uma das poucas tarefas em que os modelos de IA ainda não alcançaram resultados plenamente satisfatórios, sobretudo no que se refere à preservação rigorosa de aspectos formais como a metrifcação e a rima. Este trabalho propõe uma abordagem inovadora para enfrentar esse desafio, resultando no desenvolvimento do CarmenGen, um modelo de linguagem baseado em arquiteturas de redes neurais profundas capaz de gerar poemas que atendem a requisitos de forma. Para avaliar objetivamente os textos gerados, foi criada a métrica CARMEN, projetada para medir a conformidade com estruturas métricas e padrões de rima. Adicionalmente, foi construído o Carmen Corpus, um conjunto de poemas anotados e cuidadosamente selecionados para treinamento e avaliação do modelo, servindo como recurso valioso para futuras pesquisas na área. Os resultados obtidos demonstram que o CarmenGen atingiu desempenho médio de 91% na métrica CARMEN, cumprindo integralmente os requisitos em 95% dos casos, superando modelos de referência e estabelecendo um novo patamar para a geração automática de poemas. As contribuições deste trabalho incluem não apenas o avanço técnico na modelagem de textos poéticos, mas também o fornecimento de ferramentas de avaliação especializadas e de um *corpus* de alta qualidade. Essas contribuições ampliam as possibilidades de aplicação da IA em contextos criativos e reforçam a relevância do estudo da geração de linguagem natural como uma das áreas mais desafiadoras e promissoras da inteligência artificial contemporânea.

Palavras-chave: inteligência artificial; processamento de linguagem natural; modelo de linguagem; aprendizado profundo; geração automática de poemas

ABSTRACT

POSSI, Maurilio de Araujo, D.Sc., Universidade Federal de Viçosa, November, 2024. **Carmengen: a language model for poem generation.** Adviser: Alcione de Paiva Oliveira.

Automatic text generation by Artificial Intelligence (AI) has advanced significantly in recent decades, with models capable of producing coherent descriptions, responses, and narratives in diverse contexts. However, human language, with its structural complexity, semantic richness, and stylistic subtlety, remains an extremely challenging field of study within Natural Language Processing. Among the most complex tasks, automatic poem generation occupies a unique position, recognized as one of the few tasks in which AI models have not yet achieved fully satisfactory results, especially regarding the rigorous preservation of formal aspects such as metricality and rhyme. This work proposes an innovative approach to address this challenge, resulting in the development of CarmenGen, a language model based on deep neural network architectures capable of generating poems that meet formal requirements. To objectively evaluate the generated texts, the CARMEN metric was created, designed to measure compliance with metrical structures and rhyme patterns. Additionally, the Carmen Corpus was constructed, a set of annotated poems carefully selected for training and evaluating the model, serving as a valuable resource for future research in the field. The results demonstrate that CarmenGen achieved an average performance of 91% on the CARMEN metric, fully meeting the requirements in 95% of cases, outperforming reference models and establishing a new standard for automatic poem generation. The contributions of this work include not only technical advancement in poetic text modeling but also the provision of specialized evaluation tools and a high-quality *corpus*. These contributions expand the possibilities for applying AI in creative contexts and reinforce the relevance of studying natural language generation as one of the most challenging and promising areas of contemporary artificial intelligence.

Keywords: artificial intelligence; natural language processing; language model; deep learning; automatic poem generation

Sumário

1.	INTRODUÇÃO	9
1.1	O problema e sua importância	10
1.2	Hipótese	13
1.3	Objetivo	13
2.	REFERENCIAL TEÓRICO	14
2.1	Poema	14
2.2	Geração automática de poemas	21
3.	TRABALHOS CORRELATOS	34
3.1	<i>Large Language Models</i> atuais	37
3.2	Discussão	39
4.	CARMENGEN	40
4.1	CarmenGen, primeira versão	40
4.2	CarmenGen, segunda versão	46
4.3	CarmenGen, terceira versão	63
5.	CONSIDERAÇÕES FINAIS	74
5.1	Trabalhos Futuros	75
5.2	Publicações	75
5.3	Aspectos Éticos e Ambientais	76
	REFERÊNCIAS	77

1 Introdução

A língua natural, escrita ou falada, é a principal forma de comunicação entre os seres humanos. Desde pedidos em um restaurante até sistemas filosóficos inteiros utilizam essa forma de comunicação para transmitir conceitos, desejos, intenções, enfim, ideias. É a forma mais natural de um ser humano elaborar raciocínios e interagir com seus semelhantes. Contudo, enquanto as pessoas, em geral, têm grande capacidade em produzir e entender língua natural, podendo inclusive conseguir expressar, entender e interpretar significados bastante elaborados e sutis, essa ainda é uma tarefa extremamente complexa para as máquinas.

Esse é o objetivo da Linguística Computacional ou Processamento de Língua Natural (PLN), área da Inteligência Artificial que estuda e desenvolve métodos e algoritmos que têm como entrada ou como resultado dados em língua natural não estruturada ou semi-estruturada. É uma disciplina bastante complexa e desafiadora, dada a natureza altamente ambígua, variada, contextual e desestruturada da língua natural.

Apesar de complexa, seu desenvolvimento é de extrema importância dada a vasta gama de aplicações nas quais máquinas capazes de entender ou de se expressar em língua natural poderiam ser utilizadas, além dos benefícios que outras áreas da ciência teriam com o avanço do entendimento dos conceitos subjacentes a esta forma de comunicação humana. Por este motivo, a Ciência da Computação inclui em seus esforços o desenvolvimento de novas abordagens para tornar sistemas computacionais capazes de utilizarem recortes cada vez maiores da língua natural como protocolo de interação humano-computador, como, por exemplo, a geração de texto em língua natural que seja não só compreensível, mas também que aparente ter sido produzido por um ser humano.

A geração de texto em língua natural é um subcampo já bem estabelecido e promissor da Inteligência Artificial e da Linguística Computacional e seu principal objetivo é produzir textos que possam ser compreendidos por humanos. Dentre as pesquisas neste âmbito, destaca-se ainda a produção de textos que incluem características criativas, como narrativas, piadas, músicas e poemas (OLIVEIRA, 2009), sendo este último o foco deste trabalho.

1.1 O problema e sua importância

Os modelos de Inteligência Artificial têm se sobressaindo em diversas áreas: já geram imagens que são indistinguíveis de imagens reais; já geram pinturas melhor avaliadas do que pinturas criadas por humanos (SUN *et al.*, 2022); já criam piadas que são consideradas tão engraçadas quanto as criadas por humanos (GORENZ; SCHWARZ, 2024); já geram faces consideradas mais reais do que imagens de faces reais (NIGHTINGALE; FARID, 2022). Esse avanço não é diferente na área de geração de texto. Os *Large Language Models* (LLMs) conseguem produzir textos que já são indistinguíveis dos escritos por humanos (JAKESCH *et al.*, 2023). Contudo, alguns resultados evidenciam que os modelos de língua ainda não são capazes de gerar poemas indistinguíveis dos gerados por humanos (LINARDAKI, 2022; ELAM, 2023).

O estudo sobre a geração automática de poemas tem sido um desafio e um tópico recorrente nas pesquisas em PLN, na Linguística Computacional e em Inteligência Artificial. Na Ciência da Computação, o interesse em métodos automáticos de geração de poemas começa na década de 60, antes mesmos dos computadores se tornarem populares (OLIVEIRA H., 2017). Um poema é uma expressão artística da língua natural humana e o estudo de sua geração automática não é limitado somente à geração de textos poéticos, mas envolve o estudo de diversos níveis de representação da linguagem, como fonética, sintaxe, semântica, metrificacão, rimas, figuras de linguagem *etc.*, visando gerar textos estruturados, esteticamente agradáveis e com um certo grau de criatividade (OLIVEIRA H., 2017). Por ser uma atividade complexa e considerada exclusiva da espécie humana, sua simulação é um desafio interessante para as pesquisas em Inteligência Artificial e Linguística Computacional (LOLLER-ANDERSEN; GAMBÄCK, 2018; CRUYS, 2020; WANG *et al.*, 2021; CHAKRABARTY *et al.*, 2022a; TIAN; PENG, 2022; MOREIRA *et al.*, 2022).

Para salientar a importância do tema, recordamos que um dos precursores da Ciência da Computação, Alan Turing, destacou que a produção de poemas seria um dos grandes desafios para as máquinas. Em seu célebre trabalho “*Computing machinery and intelligence*”, Turing propôs um experimento visando identificar se “um computador pode pensar” através de perguntas e respostas. Ele nomeou seu experimento como “*The Imitation Game*”, mas o mesmo ficou mais conhecido como “Teste de Turing”. Ao demonstrar como deveria funcionar seu experimento, Turing inicia questionando o suposto computador sobre poemas: “*Please write me a sonnet on the subject of the Forth Bridge*” (TURING, 1950, p. 434) (em português, “por favor, escreva-me um soneto sobre Forth Bridge”).

Esse reconhecimento destaca a complexidade envolvida na criatividade,

na expressão artística e na compreensão da subjetividade da linguagem humana. Estes desafios continuam a inspirar pesquisas e inovações na geração automática de poemas (MOREIRA *et al.*, 2022), impulsionando não só a pesquisa em Linguística Computacional, como também aplicações práticas em áreas como: assistentes virtuais, criação de conteúdo ou até mesmo terapia digital. A habilidade de máquinas criarem expressões artísticas adiciona riqueza na interação humano-computador (CHAKRABARTY *et al.*, 2022b).

Neste sentido, já existem diversas ferramentas que utilizam modelos de linguagem¹ para auxiliar os seres humanos na produção de textos, que vão desde a autocompletação de sentenças em editores de texto ou de mensagens de correio eletrônico até sofisticados geradores de texto para auxiliar escritores, capazes de gerarem textos completos baseados em informações diversas, como resultados esportivos ou do mercado financeiro. Contudo, a criação automática de textos criativos é muito mais desafiadora, por requerer a criação de textos originais que, além de ter um sentido e estarem gramaticalmente corretos, precisam ser atraentes e ter um conteúdo lírico, o que fez com que essa tarefa fosse considerada inatingível para máquinas (KÖBIS; MOSSINK, 2021).

Os primeiros sistemas de geração de poesia baseavam-se em sua maioria em processos de recombinação de trechos retirados de um conjunto de poesias escritas por humanos. Contudo, evidenciou-se que a geração através do uso de Inteligência Artificial, além de permitir a escolha inteligente do conteúdo considerando a semântica dos termos, possibilita uma exploração mais ampla do espaço de possibilidades, o gerenciamento de mais características — muitas vezes já orientando o processo para um determinado objetivo — e a geração de poemas com um nível maior de originalidade (OLIVEIRA H., 2017).

Além disso, embora o poema possa priorizar uma estética agradável ao invés de transmitir uma mensagem clara, alcançar este último é uma tarefa altamente desafiadora para uma máquina, em especial considerando as restrições estéticas. Isso explica as diversas pesquisas na área onde diferentes abordagens são exploradas, relatadas em artigos científicos de forma detalhada e comparadas com trabalhos anteriores. É uma tarefa tão complexa quanto interessante para a pesquisa em Inteligência Artificial e Linguística Computacional (OLIVEIRA, 2009; OLIVEIRA H., 2017; LOLLER-ANDERSEN; GAMBÄCK, 2018). O domínio da técnica de se produzir texto com estilo, estética ou forma específicos pode abrir novos caminhos à geração automática de textos que sejam não só compreendidos, mas também agradáveis aos leitores, além de poder proporcionar um melhor enten-

¹ Argumenta-se que o termo correto seja “modelo de língua”, especialmente no campo da Linguística. Contudo o termo “modelo de linguagem” se consolidou na Inteligência Artificial e, portanto, foi escolhido para nomear os modelos que geram língua natural neste trabalho.

dimento dos complexos conceitos por trás da comunicação mediante língua natural.

Em que se pese os avanços dos modelos de linguagem baseados em redes neurais atuais, especialmente os LLMs, a geração automática de poemas ainda continua como um problema desafiador (ORMAZABAL *et al.*, 2022). Especula-se que isto se deve ao fato de que estes modelos são treinados em todo tipo de texto, onde poemas são minoria, o que faz com que a amostragem tenha maior probabilidade em gerar textos em forma de prosa. Isso faz com que seja necessário lançar mão de técnicas específicas a fim de se obter um modelo de linguagem especializado, capaz de produzir poemas com maior qualidade e consistência (ORMAZABAL *et al.*, 2022).

Diante do exposto, o presente trabalho apresenta um modelo de linguagem baseado em rede neural especializado na geração de poemas. O modelo é capaz de gerar poemas² em diversos formatos conhecidos, com metrificação dos versos e esquema de rimas conforme os hiper-parâmetros definidos. Além disso, no processo de criação deste modelo, foram desenvolvidos uma métrica de avaliação automática específica para poemas e um *corpus* específico de poemas com anotações. Os poemas gerados pelo modelo foram melhor avaliados do que os gerados por outros modelos estudados. Todos os resultados foram publicados em periódicos e eventos da área, contribuindo para o avanço da pesquisa científica em geração de textos em língua natural.

O trabalho está estruturado da seguinte forma: no Capítulo 2 é apresentado o referencial teórico, onde são levantados os aspectos gerais sobre poemas (seção 2.1) e sobre sua geração automática (seção 2.2); no Capítulo 3 são apresentados os trabalhos correlatos; no Capítulo 4 é apresentado o modelo proposto pelo trabalho, dividido em três versões, cada uma com seus resultados e discussão; e, por fim, no Capítulo 5 são apresentadas as considerações finais e as propostas de trabalhos futuros (seção 5.1).

² Por limitações que serão discutidas mais adiante, o modelo se restringe a gerar poemas na língua inglesa apenas.

1.2 Hipótese

É possível gerar poemas automaticamente, que apresentem características como versos metrificados e rimas, utilizando um modelo de linguagem baseado em redes neurais artificiais.

1.3 Objetivo

Propor um modelo de linguagem para geração automática de poemas. O modelo proposto deve ser executado em *hardware* com configuração modesta, tipicamente com uma única placa gráfica de uso doméstico, e mesmo assim ter um desempenho próximo ao estado da arte quanto à tarefa específica de se gerar poemas com versos metrificados e com rimas. O desempenho deve ser aferido por meio de uma métrica de avaliação específica.

Para atingir os objetivos almejados, será proposto um modelo funcional de linguagem especializado em gerar poemas. Os resultados do modelo proposto serão comparados aos resultados de outros modelos considerados estado da arte ou avaliados por uma métrica de avaliação específica para poemas.

Especificamente, a pesquisa deverá ser composta pelas seguintes etapas:

- Aprofundar os conceitos de arquiteturas de redes neurais artificiais e como podem ser aplicados à tarefa de geração de textos, especialmente poemas;
- Levantar o *baseline* e o estado da arte da área e os trabalhos correlatos que utilizem abordagens similares para serem utilizados como critérios de comparação;
- Definir os critérios que serão utilizados para avaliação do modelo;
- Investigar os diferentes conceitos apresentados por outras áreas sobre a natureza e a estrutura da linguagem, que poderão servir como inspiração para o modelo a ser proposto;
- Propor um modelo de linguagem para a tarefa de geração de poemas;
- Implementar um protótipo do modelo proposto e avaliar seus resultados, seja por uma métrica de avaliação específica ou por comparação com o estado da arte identificado;
- Contribuir com a comunidade científica publicando os resultados obtidos, bem como o modelo proposto.

2 Referencial Teórico

2.1 Poema

“ — *A Senhora de Lórien! Galadriel! — exclamou Sam. — O senhor deveria vê-la, realmente deveria, senhor. Sou apenas um hobbit, e trabalho como jardineiro em casa, senhor, se o senhor me entende, e não sou muito bom em poesia — não para compor poesia: algumas rimas cômicas, talvez, mas não poesia de verdade —, por isso não posso expressar meus sentimentos. Precisariam ser cantados.*

— **J. R. R. Tolkien**

(O Senhor dos Anéis: As Duas Torres)

O poema é uma expressão artística textual capaz de evocar estados emocionais complexos, sendo assim um reflexo da própria complexidade da experiência humana. Seu estudo remonta à antiguidade, quando Aristóteles (384 a.C.-322 a.C.) escreveu o livro “Da arte poética” onde o autor discute a importância dessa arte na consolidação das emoções e formação do pensamento civilizado (OLIVEIRA S., 2017). E para podermos contextualizar adequadamente o escopo do presente trabalho é necessário caracterizar, mesmo que de forma geral, essa forma de expressão artística do ser humano.

2.1.1 Poema vs. Poesia

Inicialmente é importante salientar a definição que será adotada para os termos “poema” e “poesia”, uma vez que esses termos podem apresentar significados distintos dependendo do contexto, do idioma e da época¹.

No contexto deste trabalho, poema é uma estrutura textual específica, com uma forma de redação definida, divididos em versos organizados em estrofes, em contraste à estrutura em prosa, que apresenta linhas contínuas organizadas em períodos que constituem parágrafos. Já o termo poesia abrange um estilo linguístico mais amplo que transcende a forma textual, caracterizado pela plurissignificação, ambiguidade e estranheza (COSTA, 2018; OLIVEIRA S., 2017). Em outras

¹ <www.etymonline.com/word/poem>

palavras, enquanto o termo poema refere-se a uma estrutura do texto, composta por versos, o termo poesia refere-se às características inerentes ao conteúdo, *i.e.*, à intenção artística do autor. O Quadro 1 ilustra um texto em forma de poema.

Quadro 1 – Exemplo de um poema.

Amor é fogo que arde sem se ver; É ferida que dói, e não se sente; É um contentamento descontente; É dor que desatina sem doer. É um não querer mais que bem querer; É um andar solitário entre a gente; É nunca contentar-se de contente; É um cuidar que se ganha em se perder. É querer estar preso por vontade; É servir a quem vence, o vencedor; É ter com quem nos mata, lealdade. Mas como causar pode seu favor Nos corações humanos amizade, Se tão contrário a si é o mesmo Amor?

Trata-se do poema lírico “Amor é fogo que arde sem se ver” do escritor português Luís de Camões (1524-1580). É possível verificar que o poema é composto por quatorze versos. Como trata-se de um soneto (do italiano, *sonetto*: pequena canção), tem ainda uma forma mais específica, *i.e.*, os quatorze versos são divididos em quatro estrofes: as duas primeiras são formadas por quatro versos (quartetos) e as duas últimas por três versos (tercetos).

Fonte: (CAMÕES, 1598)

Dessa forma, os dois termos referem-se a naturezas distintas de um texto, que normalmente ocorrem juntos, mas não necessariamente. Podemos, assim, ter um texto em formato de prosa que é poético (gênero lírico), como “Grande Sertão: Veredas” (ROSA, 1958), ilustrado no Quadro 2; e um texto em forma de poema que não seja poesia (gênero épico ou narrativo), como “Os Lusíadas” (CAMÕES, 1910), ilustrado no Quadro 3.

Sendo assim, dentre todas as características linguísticas presentes em um poema, a essencial, isto é, a que diferencia um poema de outro gênero literário, é a forma. E de todas as características que definem a forma do poema, as principais são a metrificacão e a rima (OLIVEIRA H., 2017).

Quadro 2 – Exemplo de uma prosa poética.

Eu sou é eu mesmo. Diverjo de todo o mundo... Eu quase que nada não sei. Mas desconfio de muita coisa. O senhor concedendo, eu digo: para pensar longe, sou cão mestre — o senhor solte em minha frente uma idéia ligeira, e eu rastreio essa por fundo de todos os matos, amém!

Trata-se de um trecho de “Grande Sertão: Veredas” do escritor João Guimarães Rosa (1908-1967).

Fonte: (ROSA, 1958)

Quadro 3 – Exemplo de um poema épico ou narrativo.

Assim fomos abrindo aqueles mares,
Que geração alguma não abriu,
As novas ilhas vendo e os novos ares,
Que o generoso Henrique descobriu;
De Mauritània os montes e lugares,
Terra que Anteu num tempo possuiu,
Deixando à mão esquerda; que à direita
Não há certeza doutra, mas suspeita.

Passamos a grande Ilha da Madeira,
Que do muito arvoredo assim se chama,
Das que nós povoamos, a primeira,
Mais célebre por nome que por fama:
Mas nem por ser do mundo a derradeira
Se lhe aventajam quantas Vénus ama,
Antes, sendo esta sua, se esquecerá
De Cipro, Gnido, Pafos e Citera.

O quadro ilustra um poema épico ou narrativo, também conhecido como prosa em verso. Trata-se de um trecho de “Os Lusíadas” do escritor português Luís de Camões (1524-1580). Apesar do texto estar disposto em versos metrificados, seu conteúdo não é poético.

Fonte: (CAMÕES, 1910, Canto V, estrofes 4 e 5)

2.1.2 Metrificação

O ritmo é um componente comum na experiência humana e não seria diferente quando nos referimos à linguagem, sendo importante tanto em prosa quanto em verso. Mas no poema, o ritmo possui uma relevância ainda maior, por estar intimamente relacionado à forma.

A metrificação (estudo da métrica ou medida do verso) é o mecanismo estilístico que o autor utiliza para dar ritmo ao poema, utilizando normalmente de três conceitos para alcançá-lo: a escansão, o ritmo verbal e as cesuras.

A escansão diz respeito à contagem do número de sílabas poéticas que

cada linha (ou verso) do texto apresenta. Esta forma de se medir o verso se difere da contagem de sílabas gramaticais, por dar ênfase à sonoridade (fonemas). O Quadro 4 ilustra a escansão de um verso.

O ritmo verbal está relacionado com a alternância da tonicidade das sílabas para criar a cadência do verso. Essas unidades rítmicas do poema são conhecidas como “pés”². Já as cesuras são quebras propositais deste ritmo.

Além deles, existem inúmeros outros conceitos referentes à metrificação, mas que não serão relevantes para o presente trabalho e por isso não serão abordados.

Quadro 4 – Exemplo de escansão de um verso.

1	2	3	4	5	6	7	8	9	10	x
é/	um/	con/ten/ta/men/to/				des/	con/ten/te			
1	2	3	4	5	6	7	8	9	10	x
As/	ar/mas/	e os/	ba/rões/	as/si/na/la/dos						

O primeiro verso foi extraído do poema “Amor é fogo que arde sem se ver” do escritor português Luís de Camões (1524-1580), ilustrado no Quadro 1. Já o segundo verso foi extraído da obra “Os Lusíadas”, do mesmo autor. A contagem de sílabas fonéticas termina na última sílaba tônica, além de considerar as uniões fonéticas entre palavras, como em “e os” no segundo verso. É possível verificar que os versos em questão são decassílabos.

2.1.3 Rima

Segundo (OLIVEIRA S., 2017), “rima é a repetição de sons semelhantes, ora no final dos versos, ora no seu interior, ora em posições variadas, criando um parentesco fônico entre palavras presentes em dois ou mais versos”. O Quadro 5 mostra um poema destacando as sílabas que participam da rima.

A rima é o elemento de sonoridade poética de maior relevância (OLIVEIRA S., 2017) em um poema. A repetição de sons ao longo do verso ou no seu final permite ao autor criar diversas cadências de ritmo, aumentando sua capacidade de expressão sonora. Por muito tempo, a rima era a principal critério para classificar o valor de um poema, e estava ligada também à facilidade de memorização destes (OLIVEIRA S., 2017; OBERMEIER *et al.*, 2013).

Para indicar a rima no texto, no contexto deste trabalho, adotou-se utilização de letras maiúsculas para indicar o grupo de fonemas da rima, isto é, os conjuntos de sílabas com o mesmo som. Por exemplo, no Quadro 5, a letra “A” indica os fonemas “er”, já a letra “B” indica os fonemas “ente”.

² Os poemas eram normalmente recitados com o acompanhamento de instrumentos musicais, como a lira. Daí a origem do termo “lírico” ao se referir ao gênero poético e também do termo “pé”, que era como os artistas marcavam o compasso (ritmo): com os pés.

Quadro 5 – Exemplo de esquema rímico.

Amor é fogo que arde sem se <u>ver</u> ;	(A)
É ferida que dói, e não se <u>sente</u> ;	(B)
É um contentamento <u>descontente</u> ;	(B)
É dor que desatina sem <u>doer</u> .	(A)
É um não querer mais que bem <u>querer</u> ;	(A)
É um andar solitário entre a <u>gente</u> ;	(B)
É nunca contentar-se de <u>contente</u> ;	(B)
É um cuidar que se ganha em se <u>perder</u> .	(A)
É querer estar preso por <u>vontade</u> ;	(C)
É servir a quem vence, o <u>vencedor</u> ;	(D)
É ter com quem nos mata, <u>lealdade</u> .	(C)
Mas como causar pode seu <u>favor</u>	(D)
Nos corações humanos <u>amizade</u> ,	(C)
Se tão contrário a si é o mesmo <u>Amor</u> ?	(D)

Trata-se do poema “Amor é fogo que arde sem se ver” do escritor português Luís de Camões (1524-1580). O poema foi anotado com o identificador do esquema rímico de cada verso, a saber: ABBA, ABBA, CDC, DCD. Cada letra maiúscula identifica um grupo de fonemas: A = er; B = ente; C = ade; D = or.

Fonte: (CAMÕES, 1598)

A rima pode ser classificada, quanto a sua posição no verso, como rima interna, que ocorre ao longo do mesmo verso, e rima externa, quando se dá no final de versos diferentes. Este último é o tipo mais conhecido, devido a sua sonoridade ser mais perceptível (OLIVEIRA S., 2017).

Quanto à semelhança sonora, a rima pode ser classificada como toante, quando há semelhança sonora apenas entre as vogais tônicas das palavras, e consoante ou soante, sendo a rima perfeita, quando apresenta concordância sonora entre todos os fonemas a partir da vogal tônica da palavra.

Quanto a sua localização no poema, as rimas também podem ser classificadas como:

- cruzadas ou alternadas: quando os versos rimam alternadamente, obedecendo ao esquema ABAB;
- emparelhadas, paralelas ou geminadas: rimas entre dois versos consecutivos, obedecendo ao esquema AABB;
- interpoladas, intercaladas ou opostas: aparecem no primeiro e no último verso, como em ABBA;

- misturadas: que não seguem os esquemas anteriores.

Esses dois recursos de estilo de linguagem, a metrifcação do verso e a rima, fazem com que o texto tenha uma sonoridade, que ocorre em intervalos determinados e reconhecíveis, imprimindo no texto características como ritmo e musicalidade.

Existem ainda outras classificações e detalhes sobre as rimas que, por serem irrelevantes para o presente trabalho, não foram citados.

2.1.4 Verso

Ao contrário da prosa, que possui linhas e parágrafos, o poema é dividido em versos e estrofes.

Diz-se regulares os versos dos poemas que apresentam tanto metrifcação quanto rima. Contudo, há poemas cujos versos apresentam apenas metrifcação, ditos brancos, e ainda aqueles que não apresentam metrifcação ou rima, ditos livres.

Os versos também podem ser classificados quanto ao número de sílabas poéticas. Por exemplo, os versos decassílabos têm dez sílabas poéticas, enquanto os versos dodecassílabos têm doze sílabas poéticas.

Um conjunto de versos agrupados forma uma estrofe, que pode ser classificada pelo número de versos que a compõe. Por exemplo, estrofes com três versos são chamadas de tercetos, já estrofes com quatro versos são os quartetos. Normalmente se utiliza uma linha em branco para se separar as estrofes.

2.1.5 Formas clássicas

É notório a importância da forma na identificação de um poema, tese esta corroborada pela existência de formas tradicionais, identificadas por um título e com um conjunto próprio de regras convencionadas. É o caso do soneto, como o ilustrado no Quadro 1, uma forma clássica de poema com quatorze versos, normalmente decassílabos, divididos em dois quartetos e dois tercetos. Dependendo da tradição, os sonetos podem ter diferentes agrupamentos, padrões de tonalidade e esquemas rítmicos, como ABAB CDCD EFEF GG. Há também o haikai (do japonês 俳句, *haikai*), uma forma tradicional de poemas japoneses, com dezesseite sílabas divididas em três linhas, contendo cinco, sete e cinco sílabas respectivamente (Quadro 6). Já o *limerick* (Quadro 7) é um poema de cinco linhas, normalmente com a primeira, segunda e quinta linhas mais longas, e um esquema

de rima do tipo AABBA (OLIVEIRA H., 2017). Além desses, existem ainda diversas outras formas e variações destas.

Quadro 6 – Exemplo de um poema na forma haikai.

Em cima do túmulo, cai uma folha após outra. Lágrimas também...

Trata-se do poema do autor Masuda Goga (1911-2008). Um haikai normalmente faz uma referência a uma estação do ano, chamada *kigo* (季語). Neste exemplo, é possível notar a referência ao outono.

Quanto ao verso, pode-se citar um exemplo de metrificação clássica muito conhecido, o pentâmetro iâmbico, composto por cinco iambos, o qual é uma unidade rítmica do poema formada por uma sílaba átona e uma sílaba tônica. Em outras palavras, um verso que utiliza este tipo de metrificação será um decassílabo com as sílabas tônicas ocorrendo nas sílabas pares (segunda, quarta, sexta...).

Quadro 7 – Exemplo de um poema na forma *limerick*.

Com forte estrondo, um grande galho Partiu-se no velho carvalho, Efeito do aflito Obeso mosquito Que se aboletou nesse galho!

Trata-se do poema da autora Tatiana Belinky (1919-2013).

Entretanto, apesar de ser essencial, apenas seguir uma forma reconhecida de poema não é o suficiente para um texto ser reconhecido como sendo poético ou lírico. Para tanto, o texto precisa ter um conteúdo com as características já mencionadas: plurissignificação, ambiguidade e estranheza. Pode-se dizer que, se a forma é a moldura, o conteúdo é o que ela engloba. O poeta pode utilizar a forma textual, *i.e.*, a rima e a metrificação do verso, para realçar características do texto poético, auxiliando-o a transmitir as emoções e sensações que intencionava.

Dados os conceitos básicos sobre poemas e suas formas, é necessário agora investigar como é possível gerá-los automaticamente.

2.2 Geração automática de poemas

“ *POLICEMAN: — You’re a machine. An imitation. An illusion of life. Can a robot write a symphony? Can a robot take a blank canvas and paint a masterpiece?*
ROBOT: — Can you?

— **Hillary Seitz**

(I, ROBOT, baseado no conto de Isaac Asimov)

A geração automática de poemas é um campo de pesquisa que já explorou diversas abordagens ao longo das últimas décadas. Inicialmente, os métodos mais comuns baseavam-se em regras fixas ou em processos de recombinação de trechos poéticos previamente escritos por humanos, muitas vezes organizados em estruturas pré-definidas para garantir a metrificação do verso ou a rima (OLIVEIRA H. *et al.*, 2017). Com o avanço das técnicas estatísticas, surgiram modelos baseados em n -gramas e em autômatos finitos com restrições (HOPKINS; KIELA, 2017), seguidos por estratégias que incorporaram heurísticas fonéticas e algoritmos de busca. Mais recentemente, diferentes trabalhos passaram a empregar modelos de aprendizado de máquina supervisionado, como redes LSTM especializadas em metrificação ou rima (LAU *et al.*, 2018a), e abordagens com múltiplos módulos treinados separadamente para compor estruturas poéticas complexas. Também foram propostos métodos híbridos que aplicam transformações sintéticas no *corpus* para influenciar a distribuição de probabilidade durante o treinamento (POPESCU-BELIS *et al.*, 2023). Apesar da variedade de técnicas, os melhores resultados em geração de texto reportados na literatura têm sido alcançados por modelos de linguagem baseados em redes neurais profundas, que podem apresentar maior flexibilidade e capacidade de abstração na geração de textos poéticos.

2.2.1 Arquiteturas de redes neurais profundas para geração automática de texto

A pesquisa em PLN surgiu ainda na época dos cartões perfurados (MOREIRA *et al.*, 2022), desde quando o poder computacional evoluiu exponencialmente. Esse progresso culminou com o aparecimento das arquiteturas de redes neurais com aprendizado profundo (*deep learning*) as quais mostraram resultados impressionantes em diversas áreas, como visão computacional, reconhecimento de padrões, análise de tráfego de redes, entre outras. Surgiu então uma

tendência em utilizar-se destas arquiteturas para resolução de problemas computacionais considerados complexos, o que não foi diferente na PLN, cujos resultados em estado da arte se baseiam em soluções que utilizam redes neurais artificiais com aprendizado profundo (IQBAL; QURESHI, 2020).

Como aprendizado profundo (*deep learning*) entende-se utilizar redes neurais profundas, *i.e.*, redes com camadas ocultas de neurônios, em um conjunto massivo de dados com o intuito de que a rede aprenda como resolver uma determinada tarefa (TORFI *et al.*, 2020). Várias arquiteturas de *deep learning* já foram propostas e muitas delas modificaram o estado da arte em suas tarefas. Dentre estas arquiteturas, destacam-se no PLN: *Convolutional Neural Networks* (CNNs), *Recurrent Neural Networks* (RNNs), *Autoencoders*, *Generative Adversarial Networks* (GANs) e *Transformer*.

As redes neurais convolucionais (CNNs) são arquiteturas inspiradas no córtex visual humano. São uma subclasse das redes neurais *feed-forward* e têm esse nome devido à operação matemática por trás da sua estrutura, a convolução. Nas tarefas de PLN, esse tipo de arquitetura recebe como entrada sentenças ou documentos inteiros, representados como matrizes. Cada linha dessa matriz representa uma palavra ou uma letra. São mais utilizadas em tarefas de classificação, como análise de sentimento e categorização em tópicos (TORFI *et al.*, 2020).

Já as redes neurais recorrentes (RNNs) mimetizam em sua arquitetura uma estrutura semelhante a várias redes *feed-forward* conectadas, em que a saída de uma rede torna-se a entrada da próxima. Essa arquitetura é capaz de guardar informações passadas, mimetizando assim um mecanismo de memória. Essa característica a torna muito útil para aplicações em PLN, já que pode identificar dependências entre termos que ocorreram em posições distantes no texto. Uma subclasse dessa arquitetura, muito utilizada também em PLN são as *Long Short Term Memory Networks* (LSTMs), que tentam alcançar uma característica de memória com uma magnitude ainda maior, podendo capturar dependências entre termos que ocorreram em etapas ainda mais distantes (TORFI *et al.*, 2020).

As *Autoencoders*, por sua vez, que mimetizam uma espécie de gargalo por onde a informação precisa passar e, para isso, a rede precisa aprender a como codificar essa informação, abstraindo-a de um lado, e decodificá-la do outro, reconstruindo-a. Em PLN, são muito utilizadas em tarefas de modelagem de sequência para sequência como tradução e redução de dimensionalidade (TORFI *et al.*, 2020).

As GANs, por sua vez, arquitetura apresentada por Goodfellow *et al.* (2014), são compostas pela combinação de duas redes distintas: a geradora e a discriminadora. O processo de treinamento baseia-se na estratégia Minimax, na qual a

rede discriminadora tem o objetivo de conseguir distinguir com a melhor precisão possível se os artefatos apresentados a ela são falsos (gerados) ou reais. A rede geradora, por sua vez, tenta gerar artefatos o mais próximo do real possível para tentar ludibriar a discriminadora. Nesse processo, a rede discriminadora aprende a avaliar os artefatos e a geradora aprende a gerar artefatos que pareçam reais. Após treinadas, normalmente se descarta a rede discriminadora e utiliza-se a rede geradora como produto final. Também são utilizadas com sucesso em alguns trabalhos de geração de texto (YU *et al.*, 2017; SAEED *et al.*, 2019).

Por sua vez, as redes neurais que utilizam a arquitetura *Transformer* têm alcançado os resultados em estado da arte em diversas tarefas, sendo, portanto, a arquitetura escolhida para o corrente estudo.

2.2.1.1 A arquitetura *Transformer*

O baixo desempenho das arquiteturas existentes até então em capturar dependências em sequências longas de palavras motivou a busca por mecanismos capazes de lidar com este problema, o que culminou na criação da técnica que ficou conhecida como “*attention*” (LUONG *et al.*, 2015; BAHDANAU *et al.*, 2015), que permite que o modelo foque nas partes relevantes do contexto.

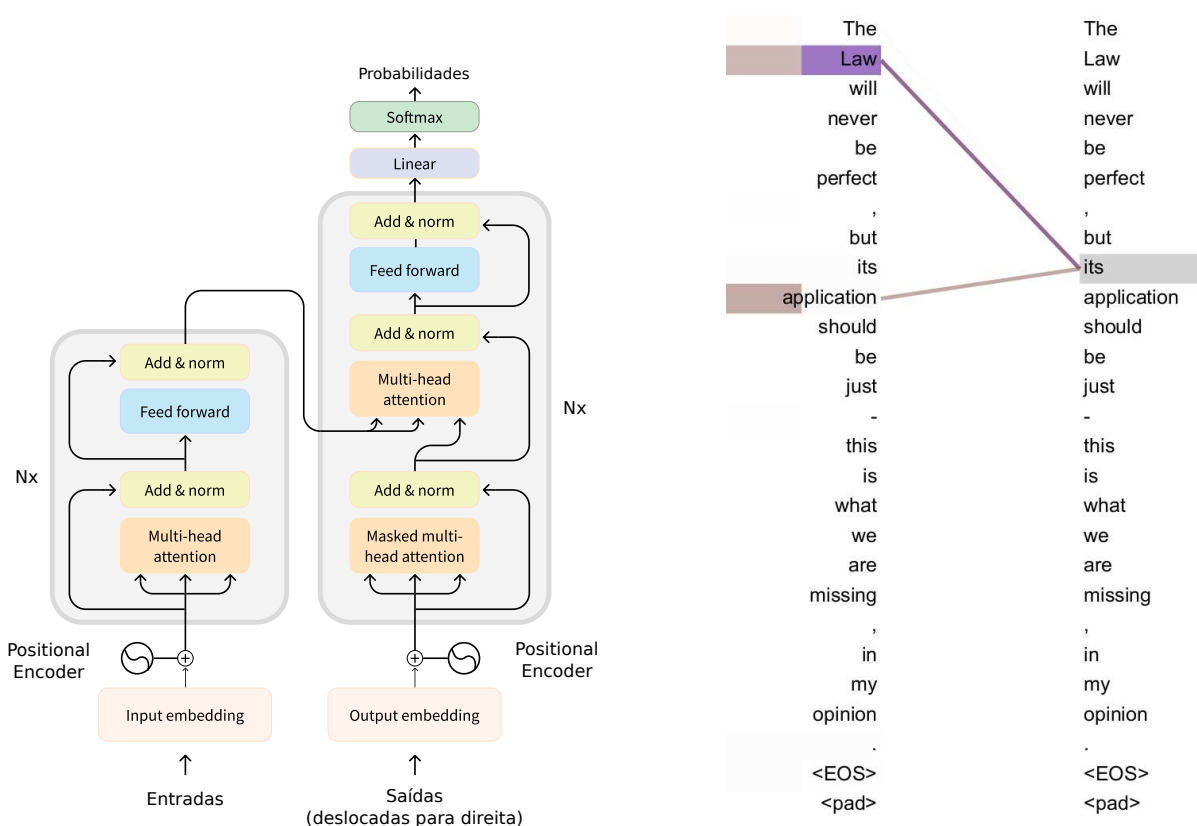
Usando uma variação da técnica de atenção, a arquitetura *Transformer* (Figura 1a), proposta por Vaswani *et al.* (2017), alcançou os melhores resultados em tarefas de PLN em diversas áreas, definindo o novo nível de estado da arte em algumas destas, como tradução e geração de textos (CELIKYLMAZ *et al.*, 2020). O sucesso da arquitetura vem, em grande parte, do mecanismo de *self attention* (Figura 1b), sem a necessidade de recorrência ou convoluções, o que permitiu que seu treinamento pudesse ser totalmente paralelizado e baseado em um número fixo de operações (VASWANI *et al.*, 2017). Consequentemente, foi possível treinar redes baseadas nessa arquitetura com um número elevado de neurônios (parâmetros) em conjuntos massivos de dados por um tempo e custo razoáveis para grandes empresas.

Contudo, este tempo e custo ainda hoje são elevados para grande parte das iniciativas científicas e uso pessoal, o que levou ao surgimento dos modelos pré-treinados. Estes modelos são desenvolvidos por setores de pesquisas de grandes empresas de tecnologia, as BigTechs, e disponibilizados, às vezes com um certo custo, aos cientistas e entusiastas do mundo todo.

2.2.1.2 Mecanismo de Atenção

Um dos componentes centrais da arquitetura *Transformer* é o mecanismo de atenção, responsável por permitir que o modelo selecione dinamicamente

Figura 1 – Detalhes da arquitetura *Transformer*.



(a) Modelo da arquitetura *Transformer*. (tradução nossa)

(b) Ilustração do mecanismo de autoatenção

A imagem à esquerda (a) mostra a arquitetura geral que engloba tanto os módulos de codificação e decodificação. A imagem à direita (b) mostra atuação do mecanismo de autoatenção (*self-attention*) enquanto foca na palavra *its*.

Fonte: (VASWANI *et al.*, 2017)

quais partes do texto devem receber mais “atenção” ao processar uma determinada entrada. Essa arquitetura utiliza uma variante, chamada *self-attention*, em que cada palavra de uma sequência presta atenção a todas as outras palavras da mesma sequência, incluindo ela mesma, para construir representações contextuais mais ricas.

Esse mecanismo é implementado como três vetores distintos: Q (*queries*, consultas), K (*keys*, chaves) e V (*values*, valores). A pontuação de atenção entre uma palavra e as demais é calculada pelo produto escalar entre sua *query* e todas as *keys* da sequência, seguido de uma normalização via função softmax. Esses pesos de atenção são então usados para combinar os valores correspondentes, produzindo a saída do bloco de atenção para cada *token* (palavra) do texto de entrada (VASWANI *et al.*, 2017). Formalmente, esse processo pode ser descrito pela Equação 2.1.

$$\text{Attention}(Q, K, V) = \text{softmax} \left(\frac{QK^\top}{\sqrt{d_k}} \right) V \quad (2.1)$$

Além disso, essa arquitetura utiliza múltiplas “cabeças” de atenção em paralelo (*multi-head attention*), permitindo que o modelo “preste atenção” a diferentes aspectos de cada relação de palavras. Esse mecanismo foi responsável por superar limitações de modelos baseados em RNNs e LSTMs, que enfrentavam dificuldades para capturar dependências longas devido à natureza sequencial do processamento (TORFI *et al.*, 2020; CELIKYILMAZ *et al.*, 2020). Além de melhorar a eficiência computacional, o uso de atenção própria permite que o modelo aprenda relações sintáticas e semânticas complexas de forma mais direta, o que é especialmente importante em tarefas como tradução automática, sumarização e geração de texto (OLIVEIRA H. *et al.*, 2017).

2.2.1.3 Treinamento: Função de perda e retropropagação

O processo de treinamento de modelos baseados na arquitetura Transformer segue o paradigma do aprendizado supervisionado. Neste cenário, a rede é apresentada a um grande conjunto de sequências de entrada, para as quais a saída esperada é conhecida, e o objetivo do treinamento é ajustar os pesos da rede de forma que ela aprenda a prever a sequência seguinte a partir de um contexto dado. Para isso, é necessário definir uma função de perda que permita avaliar o quão próximo o modelo está de sua saída desejada, bem como um algoritmo para atualizar os pesos da rede com base nesse erro.

2.2.1.3.1 Função de perda

A função de perda (\mathcal{L} , *loss function*) utilizada nos modelos de linguagem Transformer é, tipicamente, a entropia cruzada (*cross-entropy*) entre a distribuição de probabilidade predita pelo modelo para cada *token* e a distribuição real. A entropia cruzada penaliza fortemente as previsões que atribuem baixa probabilidade ao *token* correto, incentivando o modelo a aprender distribuições mais precisas. Formalmente, a função de perda pode ser definida segundo a Equação 2.2 (JURAFSKY; MARTIN, 2024).

$$\mathcal{L} = - \sum_{i=1}^N \log P(w_i | w_1, w_2, \dots, w_{i-1}) \quad (2.2)$$

Essa função força o modelo a atribuir maior probabilidade às sequências que ocorrem com frequência no *corpus* de treinamento, tornando-o mais apto a gerar textos plausíveis.

2.2.1.3.2 Gradiente descendente e retropropagação

Uma vez calculada a perda, o modelo utiliza o método de gradiente descendente para atualizar os pesos da rede neural. O princípio deste método é ajustar os parâmetros na “direção” que mais reduz o valor da função de perda, segundo o gradiente da função em relação aos pesos. Em sua forma mais simples, a atualização dos parâmetros θ segue a Equação 2.3 (JURAFSKY; MARTIN, 2024).

$$\theta^{t+1} \leftarrow \theta^t - \eta \cdot \nabla_{\theta} \mathcal{L} \quad (2.3)$$

Na Equação 2.3, η é a taxa de aprendizado (*learning rate*) e $\nabla_{\theta} \mathcal{L}$ representa o gradiente da função de perda em relação aos parâmetros da rede. Esse procedimento é realizado iterativamente para cada mini-lote de dados, utilizando variações como o Adam como otimizadores.

O cálculo dos gradientes é feito através do algoritmo de retropropagação (*backpropagation*) (VASWANI *et al.*, 2017; TORFI *et al.*, 2020), que aplica a regra da cadeia para propagar o erro da saída da rede até suas camadas iniciais. Como o Transformer é composto por múltiplas camadas empilhadas, cada bloco é treinado em conjunto, com os gradientes fluindo por toda a arquitetura. Essa capacidade de treinamento fim-a-fim é uma das razões do sucesso dos Transformers em tarefas de modelagem de sequência.

2.2.1.3.3 Normalização

Outro componente essencial no treinamento do Transformer é o uso da normalização de camada (*layer normalization*), proposta para estabilizar a distribuição das ativações dentro da rede e acelerar o treinamento. Em contraste com a normalização em lote (*batch normalization*), a normalização em camada atua sobre as ativações de cada exemplo individualmente, sendo particularmente eficaz para modelos que processam sequências variáveis em paralelo (TORFI *et al.*, 2020). Além da normalização, o Transformer também emprega mecanismos de regularização como *dropout*, aplicado nas ativações e nos pesos de atenção, com o objetivo de evitar o sobreajuste (*overfitting*).

Durante o treinamento, a estabilidade é favorecida por mecanismos como o *warm-up* do *learning rate*, em que a taxa de aprendizado é aumentada gradu-

almente nas iterações iniciais antes de começar a decair. Isso evita explosões de gradientes e acelera a convergência (VASWANI *et al.*, 2017).

2.2.2 Modelos Pré-Treinados para geração automática de texto

Até o momento, existem diversos modelos de linguagem pré-treinados que podem ser utilizados pelo público. Pode-se citar, por exemplo, o ChatGPT³, o Llama⁴, o Gemini⁵, entre outros. Estes apresentam um alto desempenho em tarefas de geração de texto. Contudo, alguns destes modelos não permitem acessar a predição probabilística de cada palavra em cada passo, outros são muito exigentes quanto às configurações mínimas de *hardware* para serem executados, e por fim, alguns exigem um investimento financeiro para sua utilização.

O *Generative Pre-trained Transformer 2* (GPT-2), por sua vez, apesar de ter sido desenvolvido em 2019, compartilha dos conceitos utilizados pelos mais avançados modelos de linguagem atualmente, como a arquitetura *Transformer* com atenção, a codificação *Byte-pair encoding* (BPE) *etc.*, apresenta um bom desempenho em gerar textos, é possível executá-lo em *hardware* mais modesto, está disponível gratuitamente e é *open source*, permitindo integrações ou alterações em sua estrutura. Essas características o tornam um excelente candidato para experimentação e observação.

2.2.2.1 *Generative Pre-trained Transformer 2*

Baseada na arquitetura *Transformer*, o modelo probabilístico de linguagem GPT-2 é uma grande rede neural profunda, com cerca de 1,5 bilhões de parâmetros, pré-treinada com base no *corpus* WebText, um conjunto massivo de dados com cerca de 40 GB de textos extraídos de 8 milhões de páginas da *web*, lançada em 2019 pela OpenAI, capaz de gerar (prever) a próxima palavra dada um conjunto inicial de palavras antecessoras (histórico ou contexto) (KÖBIS; MOS-SINK, 2021). É o sucessor do modelo GPT, tendo esta nova versão 10 vezes mais parâmetros e treinada em um conjunto de dados 10 vezes maior.

Esse modelo foi avaliado em oito tarefas distintas de linguagem natural, abrangendo uma ampla gama de habilidades linguísticas. As tarefas incluíam: (1) responder perguntas de múltipla escolha, (2) realizar inferência textual, (3) realizar análise de sentimentos, (4) responder perguntas abertas, (5) completar sentenças, (6) tradução automática, (7) sumarização de textos e (8) detecção de linguagem (RADFORD *et al.*, 2020). Em sete dessas tarefas, o modelo obteve desempenho próximo ou superior ao estado da arte da época, mesmo sem *fine-tuning*

³ <chatgpt.com>

⁴ <www.llama.com>

⁵ <gemini.google.com>

específico, apenas utilizando *zero-shot learning*, o que evidencia o potencial de generalização do modelo.

2.2.2.2 Gerando textos com GPT-2

Segundo Radford *et al.* (2019), o GPT-2 é um modelo de linguagem baseado na distribuição probabilística de sequências de palavras na forma:

$$p(w_1, \dots, w_n) \quad (2.4)$$

Considerando que a língua natural tende a ter uma ordem sequencial, ela pode ser modelada em termos da probabilidade de uma palavra ocorrer dada as suas palavras antecessoras, nesta forma:

$$p(w_n | w_1, \dots, w_{n-1}) \quad (2.5)$$

Em outras palavras, a geração de texto através do modelo GPT-2 irá propor a cada iteração a palavra que tenha maior probabilidade de acontecer dado as $n - 1$ palavras anteriores.

Este processo de geração utiliza por padrão uma estratégia gulosa para gerar os textos com base na distribuição probabilística do *corpus* aprendida pela rede neural. Neste cenário, umas das formas de se interferir no conteúdo gerado pelo modelo se faz na distribuição probabilística que será aprendida pelo mesmo mediante manipulações no *corpus*. Em modelos pré-treinados, essa manipulação se faz *a posteriori*, influenciando a distribuição outrora aprendida, em um processo chamado *fine-tuning*.

2.2.2.2.1 *Fine-tuning*

Para utilizar um modelo de linguagem pré-treinado, como o GPT-2, para tarefas específicas é possível realizar o procedimento de *fine-tuning* (RADFORD *et al.*, 2018), o qual irá realizar um treinamento supervisionado, adaptando os parâmetros da rede a um novo *corpus* específico para uma tarefa, partindo das distribuições probabilísticas aprendidas anteriormente pela rede de modo não supervisionado ou semi-supervisionado (*transfer learning*).

Vários trabalhos alcançaram relativo sucesso em suas tarefas utilizando esta abordagem, contudo, à medida que a especificidade da tarefa aumenta, também se faz mais necessário a criação de novos *corpora* e sua manipulação, e talvez a criação de um *corpus* específico para cada objetivo, o que pode tornar esse método inviável, fazendo-se necessários outros tipos de manipulação.

2.2.2.2.2 Perplexidade

Entende-se por perplexidade a métrica que quantifica o quão bem um modelo prevê uma sequência de palavras, sendo diretamente relacionada à incerteza do modelo em relação à próxima palavra a ser gerada. Em outras palavras, pode-se dizer que ela representa a “estranheza” do texto.

Formalmente, dada uma sequência de palavras w_1, w_2, \dots, w_n , e um modelo que estima a probabilidade $P(w_1, w_2, \dots, w_n)$, a perplexidade PP da sequência é definida pela Equação 2.6 (JURAFSKY; MARTIN, 2024).

$$\text{Perplexidade}(w_1^N) = \exp \left(-\frac{1}{N} \sum_{i=1}^N \log P(w_i | w_1^{i-1}) \right) \quad (2.6)$$

Ou seja, a perplexidade corresponde à exponencial da entropia cruzada (*cross-entropy*) média da distribuição real em relação à predição feita pelo modelo.

Alguns modelos permitem aumentar a perplexidade do texto a partir de um parâmetro chamado *temperature*, que será melhor discutido adiante.

2.2.2.2.3 Sampling

O processo de geração (decodificação) do GPT-2 (e também de outros modelos baseados na arquitetura *Transformer*) utiliza um método determinístico por padrão para gerar o texto, selecionando sempre a próxima palavra com base na maior probabilidade. Contudo, essa abordagem pode levar a problemas de degeneração no modelo, gerando textos sem sentido que contêm repetições indesejadas ou muito monótonos (HOLTZMAN *et al.*, 2019).

Para solucionar estas questões, existem diversas técnicas de amostragem (*sampling*) estocásticas que podem ser utilizadas. Ao utilizar uma técnica estocástica de amostragem, um fator de aleatoriedade é introduzido no processo de escolha da próxima palavra, permitindo ao modelo escolher palavras que poderiam estar “ocultas” em máximos locais. É também uma das formas de se influenciar as características do texto que será produzido pelo modelo de linguagem.

O GPT-2 implementa algumas destas técnicas, sendo possível alterar alguns hiper-parâmetros do modelo para este objetivo. Dentre estes hiper-parâmetros, podemos destacar o *temperature*, o *top-k sampling* e o *top-p (nucleus) sampling*.

O parâmetro *temperature* ($t, t > 0$), por exemplo, altera a função softmax (Equação 2.7) (JURAFSKY; MARTIN, 2024) de forma a alterar a aleatoriedade do

resultado. Quanto maior o valor de t , mais aleatório se torna o resultado (maiores as chances de se escolher uma palavra com baixa probabilidade). Por outro lado, quanto menor for t , menor a chance de uma palavra com baixa probabilidade ser escolhida. Ao se adotar o valor $t = 1$, a distribuição probabilística resultante da função será a mesma da função padrão (sem o parâmetro). Em seu limite, $t \rightarrow 0$, o processo de amostragem se comportaria de forma determinística, escolhendo sempre a palavra de maior probabilidade.

$$\text{softmax}(x_i) = \frac{e^{\frac{x_i}{t}}}{\sum_{j=1}^n e^{\frac{x_j}{t}}} \quad (2.7)$$

Já o parâmetro *top-k*, diz respeito à técnica *top-k sampling* descrita por Fan *et al.* (2018). Nesta técnica, as k palavras com maior probabilidade são filtradas e as probabilidades são redistribuídas entre elas, eliminando assim as palavras cuja probabilidade seja menor do que a probabilidade da k -ésima palavra.

O parâmetro *top-p*, por sua vez, diz respeito à técnica *top-p sampling* descrita por Holtzman *et al.* (2019). Esta técnica originou-se da observação de que utilizar um parâmetro fixo no *top-k sampling* poderia permitir a escolha de palavras com probabilidade muito baixa em distribuições mais heterogêneas, ou impedir a escolha de boas palavras candidatas em distribuições mais homogêneas. Dessa forma, Holtzman *et al.* (2019) propôs uma técnica que, ao invés de eliminar as palavras que tenham probabilidade menor que a k -ésima palavra, limita a escolha ao conjunto do menor número de palavras cuja probabilidade cumulativa seja maior ou igual à probabilidade p . Dessa forma, o número de palavras possíveis de serem escolhidas varia conforme a distribuição probabilística do conjunto original.

2.2.3 Requisitos para geração de poemas

Dado o objetivo do trabalho em propor um modelo de linguagem capaz de gerar poemas automaticamente, espera-se que estes sejam, o máximo possível, indistinguíveis dos poemas gerados por humanos. A partir disso, um dos principais desafios é encontrar uma forma de avaliar os poemas gerados.

A primeira opção seria utilizar avaliadores humanos e propor uma experiência onde esses avaliadores iriam ser expostos a poemas gerados pelo modelo e outros extraídos de *corpora* de poemas criadas por humanos, e solicitados a identificarem quais são criados por humanos ou ainda os classificar quanto a critérios específicos. Contudo, essa abordagem apresentou algumas dificuldades

que inviabilizaram sua utilização, como a obtenção de avaliadores em número significativo.

Diante disto, optou-se por utilizar outra abordagem: identificar as características de um texto que o torna reconhecível como um poema, levando o leitor a ter uma experiência estética positiva, e criar uma métrica de avaliação capaz de analisar estas características.

2.2.3.1 Forma

Os autores Obermeier *et al.* (2013) afirmam que a metrificação do verso e a rima contribuem para a percepção estética e emocional da poesia, influenciando o processamento cognitivo e, por conseguinte, o nível de apreciação estética e envolvimento emocional, concluindo que estes dois atributos têm um impacto significativo na maneira como os poemas são percebidos e apreciados, reforçando a importância desses elementos na experiência poética.

Mais ainda, os mesmos autores (OBERMEIER *et al.*, 2013) encontraram evidências de que, embora o conteúdo poético, que apresente plurissignificação, ambiguidade e estranheza, seja importante em um poema, a avaliação estética dos poemas depende principalmente das características estilísticas do texto, independentemente da semântica.

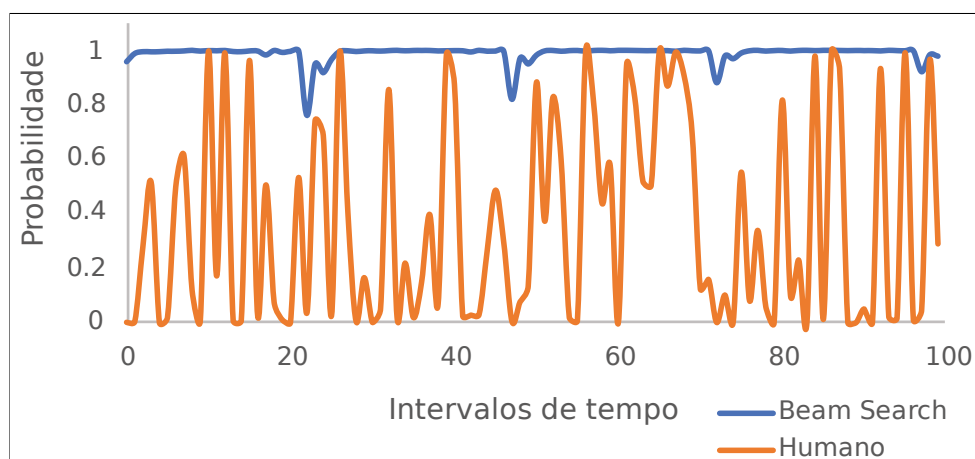
Dessa forma, se adotou estes critérios, *i.e.*, a presença de metrificação e rima no verso, como os requisitos necessários para a geração automática de poemas que possam ser reconhecidos como tais, ou indistinguíveis daqueles criados por humanos.

E para avaliar se estes requisitos serão atendidos pelo modelo, decidiu-se por utilizar uma métrica de avaliação específica para poemas, capaz de avaliá-los quanto a presença destes atributos.

2.2.3.2 Conteúdo

Outro aspecto do texto a ser considerado é seu conteúdo semântico. Além de ter a forma de um poema, é desejável que o conteúdo seja poético, *i.e.*, apresente características como plurissignificação, ambiguidade e estranheza.

Entretanto, a geração de texto por meio de modelos de linguagem, por padrão, irá propor a cada iteração a palavra que tenha maior probabilidade de acontecer dado as $n - 1$ palavras anteriores, salvo alguma variação causada pelo processo de amostragem. Esta estratégia gera textos que soam familiares e usuais, diminuindo a perplexidade (vide subseção 2.2.2.2.2) do texto gerado em relação à distribuição estatística do *corpus*.

Gráfico 1 – Comparação da perplexidade em textos.

Comparação entre a perplexidade das palavras geradas a cada intervalo de tempo pela GPT-2 usando *beam search* e a perplexidade das palavras escritas por humano

Fonte: (HOLTZMAN *et al.*, 2019, p. 2, tradução nossa)

Contudo, esta forma de produção textual acaba por gerar textos monótonos e muito previsíveis, o que, de fato, é a intenção inicial de um modelo de linguagem, mas que não seria adequado para o propósito do trabalho. Holtzman *et al.* (2019) mostra que (Gráfico 1) o texto escrito por pessoas apresenta, naturalmente, uma variação na perplexidade das palavras escolhidas. Essa perplexidade, ou estranheza, se faz ainda mais necessária em textos cuja intenção seja transmitir um conteúdo poético.

Quadro 8 – Ilustração do efeito do *fine-tuning* no conteúdo gerado pelos modelos

Love is the feeling of being loved.
Love is a thing that comes from above.

O quadro ilustra as respostas de dois modelos ao *prompt*: *What is love?*. A primeira linha apresenta a resposta do modelo que não passou pelo processo de *fine-tuning*. A segunda linha apresenta a resposta do modelo treinado em um *corpus* de poemas.

Dessa forma, além de se produzir um texto com uma forma específica, para o propósito de se gerar textos esteticamente atraentes e com um conteúdo interessante e poético, será necessário adaptar o modelo de linguagem para produzir também textos com maior perplexidade e, conseqüentemente, maior ambigüidade e plurissignificação, trazendo figuras de linguagem, metáforas, analogias, *etc.*

Entretanto, o conceito de conteúdo poético é algo difícil de se avaliar, levando em consideração que tal conceito depende do público e do contexto sócio-histórico. Neste cenário, o trabalho irá tentar alcançar um conteúdo poético atra-

vés do treinamento dos modelos em *corpora* de poemas, como ilustrado no Quadro 8, mas não irá avaliar os textos gerados considerando seu conteúdo: irá focar na forma.

Além destes requisitos, também será avaliado a consistência do modelo em gerar os textos segundo os critérios solicitados, *i.e.*, a taxa de sucesso da geração de poemas.

3 Trabalhos Correlatos

Na busca por trabalhos correlatos, foram considerados os trabalhos encontrados cujos objetivos incluíam a geração de poemas utilizando modelos de linguagem baseados em redes neurais.

O modelo proposto por Hopkins e Kiela (2017) apresentou duas metodologias diferentes para gerar poemas: a primeira baseada na utilização de *text embeddings* fonéticos e a segunda baseada em uma abordagem *adversarial* na qual a parte generativa foi composta por uma rede neural que ficou responsável por gerar o texto poético e a parte discriminativa foi constituída por uma máquina de estado finito, responsável por aceitar o texto gerado.

Na primeira abordagem, Hopkins e Kiela (2017) relataram que foi necessário treinar o modelo fonético em um *corpus* homogêneo (poesias com a mesma metrificação de verso e padrão de rima) a fim de garantir a consistência interna do modelo. A arquitetura utilizada foi a LSTM. Também relataram ter sido complexa a conversão de fonemas em palavras, dado que esta relação possui uma cardinalidade de muitos para muitos. Por fim, concluíram que a abordagem pôde aprender e gerar textos com características de metrificação e rima (vide Quadro 9), contudo, mostrou-se impraticável haja vista que seria necessário um *corpus* muito específico e um novo treinamento para cada tipo de poema a ser gerado, considerando as diversas métricas e esquemas de rimas possíveis:

“Isto não é apenas pouco prático, como também em muitos casos não existirá um *corpus* de tamanho adequado. E mesmo que existisse, um novo modelo teria que ser treinado para cada tipo de poema.” (HOPKINS; KIELA, 2017, p. 170-171, tradução própria)

Quadro 9 – Exemplo de textos gerados por Hopkins e Kiela (2017) utilizando a abordagem fonética.

<i>And humble and their fit flees are wits size but that one made and made thy step me lies</i>
<i>Cool light the golden dark in any way the birds a shade a laughter turn away</i>
<i>Then adding wastes retreating white as thine She watched what eyes are breathing awe what shine</i>
<i>But sometimes shines so covered how the beak Alone in pleasant skies no more to seek</i>

Fonte: (HOPKINS; KIELA, 2017, p. 170)

Já na segunda abordagem, Hopkins e Kiela (2017) treinaram uma rede LSTM em um *corpus* só com poesias obtidas da Internet. Contudo, como era de

se esperar, o modelo não aprendeu sobre rima ou metrificação, o que fez necessário a criação de um componente adicional, um *weighted finite state transducer* (WFST)¹ treinado em *corpora* específicos para reconhecer métricas dos versos, responsável por filtrar as sentenças geradas e aceitar somente as que atendiam aos critérios de métricas dos versos e ritmo definidos. Este método conseguiu gerar frases que atendiam aos critérios de métrica e ritmo, incluindo figuras como aliteração, consonância e assonância (vide Quadro 10). Contudo, não se observou uma regularidade nestas ocorrências em intervalos periódicos, o que se aproximaria mais do conceito de rima.

Quadro 10 – Exemplo de textos gerados por Hopkins e Kiela (2017) utilizando a abordagem com restrições WFST.

<p><i>The crow crooked on more beautiful and free, He journeyed off into the quarter sea. his radiant ribs girdled empty and very - least beautiful as dignified to see.</i></p>
<p><i>Is that people like things (are the way we to figure it out) and I thought of you reading and then is your show or you know we will finish along will you play.</i></p>
<p><i>Man with the broken blood blue glass and gold. Cheap chatter chants to be a lover do.</i></p>

Fonte: (HOPKINS; KIELA, 2017, p. 175)

O próximo modelo correlato, proposto por Lau *et al.* (2018a), é baseado em três componentes: um modelo de linguagem baseado na arquitetura LSTM com atenção, uma segunda rede LSTM com atenção responsável por aprender a métrica pentâmetro iâmbico e uma rede LSTM para aprender o conceito de rima em quartetos.

Os componentes foram treinados em *corpora* construídos especificamente para cada um. A rede de rimas, por exemplo, foi treinada em um *corpus* para avaliar, dado um par de palavras, a qualidade da rima entre eles. A geração foi limitada a quartetos, que eram gerados da seguinte forma: o modelo de linguagem gerava dez linhas que eram avaliadas quanto à qualidade da métrica pela segunda rede. A rima era imposta pela geração sucessiva de palavras finais em cada linha até que fosse obtido um conjunto de palavras que participassem de um esquema de rima. Esta estratégia acabou gerando um problema com coerência das rimas escolhidas, que foi solucionado invertendo a ordem de geração das

¹ Um WFST é um modelo matemático usado para transformar sequências de símbolos de entrada em sequências de saída, passando por estados conectados por transições com pesos numéricos.

sentenças. Desta forma, as últimas palavras de cada linha eram geradas primeiro de forma que rimassem e, em seguida, o restante das palavras de cada sentença era gerado na ordem inversa.

Os autores concluem ao final que foi possível gerar poesias com metrificação e rima, mas que mostravam problemas de legibilidade e emoção. O trabalho não mostrou exemplos dos textos gerados, embora tenha feito uma avaliação qualitativa dos mesmos.

Já o trabalho de Bena e Kalita (2019) propôs a geração de poesias utilizando GPT-2 com foco em restringir a emoção do texto gerado. Para tanto, adotou uma estratégia que envolvia realizar o *fine-tuning* do modelo GPT-2 em duas fases subsequentes: na primeira, o modelo foi treinado em um *corpus* de textos do tipo *dream-like*² em primeira pessoa para, em seguida, ser treinado novamente em um *corpus* constituído de poemas com uma emoção específica. Ao final, se obteve cinco modelos GPT-2 treinados em cada emoção: alegria, expectativa, raiva, tristeza e confiança. O Quadro 11 ilustra alguns exemplos dos poemas gerados por este trabalho.

Quadro 11 – Exemplo de textos gerados por Bena e Kalita (2019) utilizando o modelo GPT-2 com *fine-tuning* em *corpus* específicos para cada emoção.

<i>Heard I a song of joy, A song of happy sound, Fills all the air I breathe, To him I sing, to him I sing the happy song. All night long on the steep green grass I ride and sing</i>
<i>The other, who with one accord Wrote my essay, in that he was dear And good, and knew well, how we ought to treat A man of such renown, and such love? He's a good honest man, no doubt</i>

Fonte: (BENA; KALITA, 2019, p. 6)

Por fim, em Popescu-Belis *et al.* (2023) é apresentado um modelo de linguagem baseado no GPT-2 capaz de gerar versos com o esquema rímico AABB presente em 60% das amostras. Neste trabalho, os autores utilizaram um algoritmo para criar um *corpus* de versos rimantes no padrão desejado e depois realizaram o *fine-tuning* do modelo pré-treinado GPT-2 no *corpus* sintético.

O processo de criação do *corpus* sintético inclui a utilização de um dicionário de rimas, a geração de poemas com versos que não rimam, seguida da

² Textos narrativos em primeira pessoa que apresentam características associadas à experiência onírica, isto é, semelhantes aos sonhos.

Quadro 12 – Exemplo de textos gerados por Popescu-Belis *et al.* (2023) utilizando o modelo GPT-2 com *fine-tuning* em *corpus* sintético de rimas.

<i>The prince of men in arms he heard</i>
<i>So bold, so bold the warrior plundered</i>
<i>That she herself in sorrow cried</i>
<i>My God! who made the earth so bide</i>
<i>She sees no other sun above</i>
<i>Nor in that cloudless sky doth dove</i>
<i>To the sound of your sweet voice</i>
<i>As of a little bird at choice</i>

Fonte: (POPESCU-BELIS *et al.*, 2023)

substituição das últimas palavras por outras que rimem, utilizando o GPT-2 para escolher, dentre as rimas disponíveis no dicionário, a com maior probabilidade. O Quadro 12 ilustra alguns exemplos deste *corpus* sintético de versos que rimam.

Também realizaram o mesmo procedimento para o esquema rímico ABAB, contudo obtiveram um resultado pior: apenas 45% das amostras apresentavam rimas neste esquema, concluindo os autores que a distância (no texto) da ocorrência das palavras que rimam dificulta o aprendizado pelo modelo.

Para comparação, o modelo treinado no *corpus* sem nenhuma intervenção apresentou 11% de versos com rimas consecutivas.

O trabalho também apresenta uma métrica de avaliação de rimas para poemas utilizada para gerar o *corpus* sintético. Segundo o trabalho, a métrica de avaliação é capaz de dizer se dois versos rimam comparando, para isso, suas últimas palavras.

3.1 *Large Language Models* atuais

É fato notório a grande habilidade que os modelos de língua demonstram em diversas tarefas. Contudo, pode não ser tão evidente que eles ainda possuem deficiências quando o assunto é gerar poemas (ORMAZABAL *et al.*, 2022). Embora haja alguns modelos que vêm ganhando destaque, será considerado nesta discussão de trabalhos correlatos o modelo que tem atingido os melhores resultados na geração de textos: o Chat GPT.

3.1.1 Chat GPT

No Quadro 13 é ilustrado um poema produzido pelo ChatGPT³, versão 3.5, em que se pede que seja gerado um poema com quatro versos decassílabos e esquema de rima ABAB, sobre o tema “amor”. Embora ele tenha sido bem-sucedido

³ Disponível em: <<https://openai.com/chatgpt/>>. Acesso em: 23.mai.2024.

Quadro 13 – Exemplos de dois poemas gerados pelo ChatGPT v3.5.

Gere um poema de quatro versos decassílabos, com esquema de rima ABAB, sobre o tema “amor”.

No cenário do peito, o amor desperta (A)
Em versos de paixão, a alma se oferta (B)
Em cada batida, a chama se espalha (A)
Amor, poesia viva que o coração canta (B)

Faça um poema cujos versos rimem no esquema ABBA.

Sob o céu azul, a brisa dança (A)
No jardim, as flores sussurram canções (B)
A vida, em seus giros, a esperança avança (B)
Em cada verso, ecoa mil emoções (A)

No primeiro caso, a instrução é para a geração de rimas no esquema “ABAB”, contudo o modelo gerou no esquema “AABX” (considerando que “espalha” e “canta” não rimam). No segundo exemplo, o comando pedia que fosse gerado um poema com o esquema de rima “ABBA”, no entanto o modelo gerou um poema no esquema “ABAB”. Esses dois exemplos ilustram como a geração de poemas ainda é desafiador mesmo para os modelos mais avançados.

Fonte: ChatGPT v3.5

em gerar quatro versos e aderir ao tema proposto, ele falha em gerar a metrificação e a rima, embora chegue a mostrar, incorretamente, o padrão de rima de cada verso.

Isto provavelmente se deve ao fato da fonética das palavras serem ainda variáveis latentes à codificação utilizada para representar os símbolos textuais de tal forma que, embora influenciem na grafia destas, não são percebidas pelo modelo.

A codificação BPE, que se tornou o padrão *de facto* atualmente, utilizada pelos modelos de linguagem como GPT-2, GPT-3, ChatGPT, RoBERTa *etc.* (com pequenas variações entre cada um), realiza uma codificação baseada em frequência que, de um lado, possibilita que mais *tokens* sejam acomodados na janela de contexto do modelo, mas por outro gera alguns inconvenientes quando se trata de mapear características que só são percebidas ao nível de caractere ou fonético (PROVILKOV *et al.*, 2020).

No vocabulário do BPE utilizado no GPT-2, por exemplo, existem tanto *tokens* que representam uma letra, como que representam “palavras” como “raw-downloadcloneembedreportprint” e “BuyableInstoreAndOnline” (PETROV *et al.*, 2023), e nem sempre esses *tokens* serão utilizados consistentemente: o número “10”, por exemplo, pode ser codificado como o *token* 10 ou pelos *tokens* 1 e 0, ambos os casos podendo ainda serem representados por um *token* com um

caractere de espaço em branco prefixado.

Desta forma, estes modelos de linguagem podem não ver os caracteres em si, mas trechos de palavras, que podem variar muito dependendo do contexto, da palavra sendo codificada *etc.*, dificultando, ou até inviabilizando, que o modelo aprenda certos aspectos da linguagem ligados a sua pronúncia, como a rima e a metrificação⁴, já que trechos das palavras, como suas terminações rimantes, podem ser representados de maneiras completamente distintas, mesmo tendo a mesma grafia.

3.2 Discussão

Todos os trabalhos supracitados tiveram relevantes contribuições. Contudo, na busca de se gerar textos que apresentassem as restrições desejadas, versos metrificados e rimas, precisaram escolher entre utilizar abordagens complexas para conseguir resultados consistentes ou relaxar os critérios para alcançar o objetivo com uma abordagem simples, sob pena de não ter consistência nos resultados.

Um problema recorrente nos trabalhos foi a necessidade de ter um *corpus* específico — e consequentemente um modelo de linguagem específico — para cada tipo de poema (HOPKINS; KIELA, 2017), para cada tipo de emoção (BENA; KALITA, 2019) ou para cada esquema rímico (POPESCU-BELIS *et al.*, 2023). Isto gera uma limitação no alcance dos modelos, uma vez que os restringe a formas ou conteúdos muito específicos, impedindo sua generalização.

Mesmo os modelos de linguagem mais avançados, como os LLMs, ainda encontram desafios na geração de poemas com metrificação e rimas consistentes. A hipótese é que isso se deva à limitação do *embedding* BPE que utilizam ou por utilizarem vetores semânticos, o que torna difícil para esses modelos lidarem diretamente com a grafia das palavras.

Desta forma, resta evidenciada a necessidade de se prosseguir na investigação deste tema.

⁴ BRANWEN, Gwern. GPT-3 Creative Fiction. 2023. Disponível em: <<https://gwern.net/gpt-3>>. Acessado em: 23.mai.2024.

4 CarmenGen

Nesta seção serão apresentadas três versões do modelo de linguagem para geração de poemas proposto, o CarmenGen, mostrando a evolução no desenvolvimento do mesmo. Em cada uma destas seções há a apresentação da versão do modelo, seus resultados e posterior discussão.

4.1 CarmenGen, primeira versão

A primeira versão do modelo baseou-se na abordagem *transfer learning*. Nesta abordagem, utiliza-se como base um modelo pré-treinado, normalmente em um volume massivo de dados por uma estrutura de grande poder computacional e custo energético. Neste modelo é realizado então o *fine-tuning* para adaptá-lo para uma tarefa específica que, no objeto de estudo, é produzir poemas. Desta forma, ao permitir a reutilização de modelos pré-treinados, a abordagem *transfer learning* permite a utilização de modelos treinados com uma estrutura a qual o acesso é muito limitado, ao mesmo tempo que reduz o impacto ambiental da mesma (RODRIGUES *et al.*, 2022).

O modelo escolhido como ponto de partida foi o GPT-2 (subseção 2.2.2.1). Essa escolha se deu pelo fato de, à época, ser o LLM mais acessível em questão de recursos computacionais necessários para utilizá-lo, mas que já apresentava ótimos resultados, ultrapassando os níveis de estado da arte em 7 das 8 tarefas às quais se propôs (RADFORD *et al.*, 2020).

A implementação foi realizada na linguagem de programação Python, versão 3.10.12, utilizando as bibliotecas `transformers`¹, versão 4.41.2, e `accelerate`, versão 0.31.0.

A partir disso, buscou-se por *corpora* de poemas que pudessem ser utilizados para realizar o *fine-tuning* do modelo, listados a seguir.

4.1.1 *Corpora* de poemas

Embora haja inúmeros *corpora*, buscou-se aqueles que reunissem apenas poemas, no intuito de conseguir expor o modelo de linguagem a essa forma textual. Como nenhum *corpus* atendeu a este critério plenamente, admitiu-se *corpora* de textos poéticos, em geral, mas cujo conteúdo contivesse também poemas.

¹ Disponível em: <<https://huggingface.co/docs/transformers>>.

4.1.1.1 Gutenberg Poetry Corpus

O *corpus* Gutenberg Poetry² é composto de aproximadamente três milhões de linhas de poesias extraídas de centenas de livros do Projeto Gutenberg³. O *corpus* não possui anotações e, em sua maioria, é composto por textos poéticos em prosa, *i.e.*, nem sempre em forma de poemas.

4.1.1.2 Poetry Foundation Corpus

O *corpus* Poetry Foundation Poems⁴ aglutina poesias publicadas no sítio Poetry Foundation⁵, com informações sobre o autor, o título da poesia, o conteúdo e algumas *tags* retiradas do próprio sítio. Este *corpus* possui cerca de 1.324 poesias únicas, disponibilizadas em um arquivo de texto com 9 MB, no formato CSV. Ao analisarmos o *corpus*, notamos que o mesmo possui alguns textos em prosa, outros em versos, mas sem rimas. Além disso, vários textos não possuem anotações ou, quando possuem, estas anotações dizem respeito ao tema do texto e não à forma (métrica ou rima) em si.

4.1.1.3 Poems Dataset

O *corpus* Poems Dataset (NLP)⁶ é composto por cerca de 20 mil textos poéticos individualizados, sem anotações ou marcações de qualquer espécie. Os textos são apresentados em dois conjuntos: no primeiro, os textos são classificados por forma, e no segundo, agrupados por temas. No total, o *corpus* conta com 20.657 exemplares (6.295 no conjunto classificado por formas e 14.362 classificados por temas). Os textos foram extraídos do portal Poem Hunter⁷.

4.1.2 Treinamento do modelo

Definidos os *corpora* candidatos, iniciou-se o treinamento do modelo. Decidiu-se por treinar um modelo para cada *corpus*, de forma a poder comparar os resultados alcançados.

Foi realizado então o *fine-tuning* do modelo de linguagem GPT-2 para cada *corpus*. Para o treinamento foi utilizado a plataforma Google Colab, com uma instância A100. O processo de 10 *epochs* durou cerca de 12 horas para cada modelo.

² Disponível em: <github.com/aparrish/gutenberg-poetry-corpus>. Acesso em: 14.dez.2023

³ Disponível em: <[gutenberg.org](https://www.gutenberg.org)>. Acesso em: 14.dez.2023

⁴ Disponível em: <www.kaggle.com/datasets/tgdivy/poetry-foundation-poems>. Acesso em: 14.dez.2023

⁵ <www.poetryfoundation.org>

⁶ Disponível em: <www.kaggle.com/datasets/michaelarman/poemsdataset>. Acesso em: 08.dez.2023.

⁷ Disponível em: <poemhunter.com>. Acesso em: 08.dez.2023.

4.1.3 Resultados

Após o treinamento ser concluído, foram geradas diversas amostras utilizando os modelos. Foram experimentados vários parâmetros que influenciam no processo de *sampling* do mesmo, como *temperature* (t), *top k* (k), *top p* (p) *etc.*, além de variados *prompts*. O objetivo dessa avaliação empírica foi encontrar os parâmetros que maximizem o desempenho de cada modelo para poder comparar os melhores resultados de cada um, uma vez que os resultados se mostraram sensíveis aos parâmetros.

A seguir serão apresentados os resultados obtidos com o treinamento dos modelos para cada *corpus* elencado.

4.1.3.1 Modelo GPTrovador: Gutenberg

Inicialmente tentou-se *prompts* no formato de comando. O Quadro 14 ilustra um exemplo de saída do modelo. É possível observar que o modelo gerou um texto em prosa, que embora apresente certo conteúdo poético — talvez devido ao parâmetro *temperature* mais elevado, gerando maior estranheza (perplexidade), uma das características dos textos poéticos — não se parece com um poema, como discutido em subseção 2.1.1.

Quadro 14 – Amostra de um texto gerado pelo modelo GPTrovador: Gutenberg

```
We all have to give credit where it's due. When you say to me that "Love is in
my heart and my soul" is so true, I mean that, if a woman is not capable of giving
more than just a hint about her true state of mind, then my daughter will be more
than an empty toy. She will be an open heart that will always come home to me and
will tell me her own secrets and be there for me in every room I live in. Love is
not a tool to be used. It's always a tool.

Love is a gift. It's not meant to be a job. It's meant to be a kind of human need.
It's a gift to me. I have a gift of love for you.

It's a gift that makes you happy and to love.

Love is a gift that keeps us alive in every part of the world. It's an opportunity
to share our love and to be with you forever. It's a gift that brings you back to
the right place. It's a gift that makes you love and love again.

Love is a gift. It's not meant to be a job. It's meant to be a kind of human need.
It's a gift to me. I have a gift of love for you. Love is a gift. It's not meant to
be a job. It's meant to be a kind of human need. It's a gift to me.

Love is a gift that keeps us alive in every part of the world. It's an opportunity
to share our love and to be with you forever. It's a gift that brings you back to
the right place. It's a gift that makes you love and love again. Love is a gift.
It's not meant to be a job. It's meant to be a kind of human need. It's a gift to
me. I have a gift of love for you.

[...]
```

O *prompt* utilizado foi “*Write me a poem about love.*”, com os seguintes parâmetros: $t = 2,00$; $k = 50,00$; $p = 0,90$.

A seguir, foram realizados testes com *prompts* no formato de título do poema e de trecho inicial do poema. Esses experimentos mostraram um resultado mais próximo do esperado, com um texto separado em versos, como pode ser visto no Quadro 15 e no Quadro 16, mas ainda sem métrica ou rima, e degenerando facilmente para prosa à medida que a amostragem se distanciava do *prompt*.

Quadro 15 – Amostra de um poema gerado pelo modelo GPTrovador: Gutenberg

Shall I compare thee to a summer's day?
Thou art more lovely and more temperate:
Rough winds do shake the darling buds of May,
And summer's lease hath all too short a date:
 And, lo, herr me, you see of her to-night.
 Thou art most amiable.
 She's a sweet girl.
 Well, it's time to retire;
 You don't have any problems now,
 Whip me to him in her song,
 And you might at least write to her when the wind begins.

O *prompt* utilizado está enfatizado no texto (primeiras 4 linhas). Foram utilizados os seguintes parâmetros: $t = 1,00$; $k = 150,00$; $p = 0,90$.

Quadro 16 – Amostra de um poema gerado pelo modelo GPTrovador: Gutenberg

My mistress'eyes are nothing like the sun;
Coral is far more red than her lips'red;
 The firewood glows, burning with incense.
 So my master not a little troubled the night;
 Dow me an hour with the rags and dung,
 Lift me up on my loins and fill the face;
 'Twould I feel my knickers cut loose
 When I lay her on guard.
 Twould my lovers and the sea shall not dine together
 But laugh together, for all that day is and all night;

O *prompt* utilizado está enfatizado no texto (primeiras 2 linhas). Foram utilizados os seguintes parâmetros: $t = 1,00$; $k = 150,00$; $p = 1,00$. Neste exemplo, foi adicionado uma penalidade por repetição de 4-gram.

Os resultados com este modelo apontam para a necessidade de um *corpus* mais específico de poemas, que exponha a rede a mais exemplos de texto formatado em versos com rimas e conteúdo poético.

4.1.3.2 Modelo GPTrovador: Poetry Foundation

Já com o modelo treinado utilizando o *corpus* Poetry Foundation, os resultados evidenciam uma tendência maior de geração de versos e um conteúdo mais poético, com a presença de algumas figuras de linguagem próprias deste estilo. Contudo, é possível notar (Quadros 17 e 18) que os versos ainda apresentam muita irregularidade e não há um esquema de rima reconhecível.

Quadro 17 – Amostra de um poema gerado pelo modelo GPTrovador: Poetry Foundation

My mistress'eyes are nothing like the sun;
Coral is far more red than her lips'red;
 All things live in their own color.
 I could not conquer, but conquer the light too;
 Unconquer the fear that had strayed out of me,
 And cast my love's influence upon the stars.
 I will conquer the sea with a hundred lives;
 I will drown with a hundred seas;
 Oh, let me come down from the dead sea!
 I will burn on the fires of heaven with the breath
 Of death's burning breath;
 I'll bless the stars with their own delight;
 I'm not deceived by any life that's false:
 I'll sit looking out on the night
 Before the sun and hear its reflection in the dawn.
 I'll fold myself in another's arms like flowers in mine,
 And kiss the night in the kiss of its own sweet perfume,
 Giving my soul the strength it craves and fears.
 Oh, the dead eyes come out on us with a faint, strange air of glory which lies
 In its own eternal glory that we are dead.
 O, oh the breath of death's terrible things,—death's face that pulls in my heart
 to me like flame!
 The breath of souls'and life's darkenings on us! I'll burn on that will scatter
 you as a rain,—like roses in me!
 [...]

O *prompt* utilizado está enfatizado no texto (primeiras 2 linhas). Foram utilizados os seguintes parâmetros: $t = 1,00$; $k = 150,00$; $p = 1,00$. Neste exemplo, foi adicionado uma penalidade por repetição de 4-gram.

Quadro 18 – Amostra de um poema gerado pelo modelo GPTrovador: Poetry Foundation

That time of year thou mayst in me behold
When yellow leaves, or none, or few, do hang
 "Round my feet, and with one short cry
 Make a fearful meadow for me
 Where not one grave, not one grave in sight
 Shall bear the sorrow I bear. Let those
 Who have looked behind, in their homes
 Turtle howls and bugles run to hear
 Thy merry song that shall be no more heard.
 "Since the time of the dove is come,
 When the sunbeams were shed on the deep,
 The nameless maids asleep in the bowers,
 And the morn was sweet, and the dawn was golden,
 Each child in his wanderings found pleasure and fear,
 No more than some a charm unto the child
 Who trod, alone and alone on his father's good will.
 But thus—though the children have all return'd—they call
 "The rest, the all the happy day;"or at least some words
 "At last, come,"or of the one word that is still desired, or once
 The word thus with a sound, the day of "the mind has brought
 And is come again and gone."
 [...]

O *prompt* utilizado está enfatizado no texto (primeiras 2 linhas em **italico**). Foram utilizados os seguintes parâmetros: $t = 1,00$; $k = 150,00$; $p = 1,00$. Neste exemplo, foi adicionado uma penalidade por repetição de 4-gram.

4.1.3.3 Modelo GPTrovador: Poems Dataset

Por fim, os resultados do modelo treinado utilizando o *corpus* Poems Dataset mostraram os resultados mais próximos de poemas com versos metrificados e rimas. Os textos gerados apresentaram versos mais regulares e um conteúdo poético bem evidente, repleto de figuras de linguagem e outros recursos estilísticos. Embora ainda não haja geração consistente de rimas, é possível verificá-las em alguns versos, como ilustrado no Quadro 19. Neste modelo, verificou-se empiricamente que os resultados foram melhores utilizando o *prompt* no formato de comando.

Quadro 19 – Amostra de um poema gerado pelo modelo GPTrovador: Poems Dataset

Write me a poem about time.

Time is my friend and
I can share the day with you.
With a smile on your face
And a smile on your lips.
I can always tell you
when the sun is shining,
when the clouds are gathering.
I can't tell you my
life in the clouds,
when the stars are shining
and the sea is calm.
And even now when the sun is shining
and the stars are gathering,
it's too late.
Time is my friend and
I can share the day with you.
I'm always on my feet
and can always tell you
when the sun is shining
and the skies are streaming.
I'm always on my feet
when the clouds are gathering
and the birds sing their lays.
Even now when the sun is shining
and the stars are gathering,
I can't tell you when the sun is shining.
Because the moon hides her face
and the birds sing their lays,
I can't tell you when the sun is shining,
because the moon hides her face
and the birds sing their lays.
Because the sun hides her face
and the moon hides her face,
I'm always on my feet
and can't hear the birds singing,
because the moon hides her face
and the birds sing their lays
[...]

O *prompt* utilizado está enfatizado no texto (primeira linha em itálico). Foram utilizados os seguintes parâmetros: $t = 1,00$; $k = 150,00$; $p = 1,00$;

4.1.4 Discussão

Os resultados foram promissores, em especial no último modelo treinado (subseção 4.1.3.3), aproximando-se de um poema com versos metrificados e com rima, já se aproximando do resultado dos trabalhos correlatos encontrados.

Contudo, apesar dos resultados relativamente positivos, os textos gerados ainda não apresentam características como metrificação e rimas de forma consistente. Acredita-se que o motivo seja que os *corpora* utilizados contenham textos que não apresentam estas características, ou mesmo os que as apresentam, o façam de forma muito heterogênea (diferentes tipos de formas e esquemas de rimas), dificultando para o modelo capturar tais sutilezas.

4.2 CarmenGen, segunda versão

Os resultados da primeira versão mostraram evidências de que os exemplos necessários para treinar os modelos estão sub-representados nos *corpora* utilizados, o que influenciou negativamente os modelos de linguagem a produzirem poemas, uma vez que os textos gerados apresentavam inconsistências, como a presença de textos em versos e prosa juntos, versos de tamanhos muito diferentes, rimas escassas, *etc.*

A partir deste resultado, levantou-se a hipótese que um modelo de linguagem treinado em um *corpus* específico com somente poemas poderia ter um desempenho superior. Desta forma, iniciou-se a construção da segunda versão do CarmenGen, baseada no treinamento de um modelo pré-treinado em um *corpus* específico de poemas.

Para atingir este objetivo foi necessário construir tal *corpus*, pois não foi encontrado outro com estas características. No entanto, decidiu-se que seria mais vantajoso construir este *corpus* a partir do refinamento, preferencialmente automático, de outros *corpora* do que tentar criar um totalmente novo, que envolveria varrer sítios de poemas na Internet em busca de poemas que, no final, teriam que ser refinados de toda forma.

4.2.1 Avaliação automática de poemas

Embora haja diversos trabalhos visando gerar poemas, ainda há poucas iniciativas de criar métodos automáticos para avaliar a qualidade dos poemas gerados. Em grande parte, isto é devido ao grande número de diferentes formas e estilos que podem ser utilizados na criação de poemas, tornando complexo o trabalho de criar um método automático para avaliá-los, fazendo com que os pes-

quisadores recorram a outras métricas de avaliação para executarem esta tarefa. Ocorre que estas métricas de avaliação, muitas das vezes, não foram projetadas para este propósito, analisando outros critérios que não aqueles relativos aos aspectos estéticos inerentes à forma do texto (POSSI *et al.*, 2023).

Nesta seção será apresentada a proposta de uma métrica de avaliação automática de poemas, com foco nos critérios da forma do texto, *i.e.*, metrificação dos versos e esquema rímico. O intuito da criação desta métrica de avaliação, além de avaliar os poemas gerados pelos modelos de linguagem, é poder avaliar os *corpora* de poemas, permitindo a sua seleção ou melhoramento.

4.2.1.1 Métricas de avaliação de poemas

Como mencionado, na ausência de uma alternativa melhor, muitos autores escolhem utilizar métricas de avaliação como a BLUE (PAPINENI *et al.*, 2002), que fora projetada para avaliação de tradução, como métrica de avaliação para poemas. E, embora em alguns trabalhos haja algumas iniciativas de métricas de avaliação acessórias aos seus objetivos principais, não foi encontrado na literatura trabalhos cujo objetivo principal fosse a elaboração de uma métrica de avaliação específica para poemas.

A grande maioria das iniciativas apuradas baseavam-se em processos que envolviam avaliadores humanos (MANURUNG, 2004; MISZTAL; INDURKHYA, 2014; DAS; GAMBÄCK, 2014; YAN, 2016). Outros trabalhos já aplicam o Teste de Turing buscando identificar se avaliadores humanos podem distinguir entre poemas gerados pelo modelo de poemas escritos por outros humanos (RASHEL; MANURUNG, 2014). Estes métodos normalmente avaliam aspectos qualitativos, e até subjetivos, do texto, que podem ter mais relação com o conteúdo lírico do mesmo do que com sua forma, além de limitarem a sua aplicação em conjuntos massivos de dados, pela necessidade da avaliação manual.

Por outro lado, há poucos trabalhos que abordam um método automático de avaliação de poemas na literatura e, mesmo estes, usualmente baseiam-se em métricas de avaliação projetadas para outros objetivos, como sumarização ou tradução (OLIVEIRA H., 2017).

A métrica de avaliação ROUGE (LIN; OCH, 2004), acrônimo de *Recall-Oriented Undestudy for Gisting Evaluation*, projetada para avaliar a qualidade de resumos em comparação com outros criados por humanos e considerados ideais. Esta métrica de avaliação é utilizada em alguns trabalhos para avaliar poemas (YAN *et al.*, 2013; OLIVEIRA H. *et al.*, 2017).

Já a métrica de avaliação BLUE (PAPINENI *et al.*, 2002), acrônimo de *BiLingual Evaluation Understudy*, foi projetada para substituir avaliadores humanos

em processos de avaliação de traduções. Também recorre à metodologia de se comparar as traduções que se quer avaliar com um *corpus* de traduções consideradas padrão ouro. Esta métrica de avaliação também é utilizada para analisar poemas em alguns trabalhos (ZHANG; LAPATA, 2014; YAN, 2016).

Vale mencionar outros métodos automáticos de avaliação encontrados, como comparar a perplexidade de poemas gerados automaticamente com aqueles gerados por humanos (YAN *et al.*, 2013; YAN, 2016) e estimar a similaridade semântica usando a média da similaridade do cosseno entre textos gerados automaticamente e aqueles gerados por humanos (WONG *et al.*, 2008; POTASH *et al.*, 2015).

Todas essas iniciativas de avaliação mencionadas, apesar de suas contribuições, acabam envolvendo avaliadores humanos ou utilizando métricas de avaliação que não foram projetadas para este fim, uma vez que focam mais no conteúdo dos textos e não em sua forma.

Dentre os trabalhos que apresentam métricas de avaliação, mesmo que de forma acessória, que poderiam ser utilizadas para poemas, destacam-se Malmi *et al.* (2016), onde o autor apresenta uma métrica de avaliação de letras de rap, mas que visa a avaliar a quantidade de rimas presentes nos versos, conceito que o autor chamou de “densidade de rima”, avaliando a qualidade das letras geradas comparando a densidade de rimas destas com a densidade de rimas presente em letras de músicas criadas por humanos; e Popescu-Belis *et al.* (2023), que apresenta uma metodologia para contar a quantidade de versos que rimam em um poema, ou avaliar se dados dois versos rimam. Contudo, os dois trabalhos focam na avaliação de rimas, não considerando a questão da metrificação dos versos.

4.2.1.2 A métrica de avaliação Carmen

Dado esta lacuna na literatura de uma métrica de avaliação automática específica para poemas, decidiu-se por propor a CARMEN (do latim, *carmen metimur*: medir o poema). É uma métrica de avaliação específica para este propósito, capaz de avaliar quantitativamente características morfológicas do texto, como a metrificação dos versos e o esquema rímico, em diferentes tipos de poemas, como sonetos, limericks, haicais *etc.*, podendo ser utilizada tanto para o treinamento supervisionado de modelos, quanto para a avaliação de *corpora* ou poemas isolados (POSSI *et al.*, 2023).

Esta métrica de avaliação foi projetada com dois modos de operação: CARMEN-d (definida), usada quando se quer avaliar poemas sabendo *a priori* qual a metrificação ou o esquema rímico que os mesmos deveriam ter; e a CARMEN-i (indefinida ou por inferência), quando o formato do poema sendo avaliado não

é conhecido de antemão e caberá à métrica de avaliação tentar identificá-lo.

Em (POSSI *et al.*, 2023), os autores demonstraram seu poder discriminatório aplicando-a a diversos textos que continham exemplos tanto de prosas, textos poéticos e poemas, e a métrica CARMEN foi capaz de separar os poemas dos demais textos com sucesso.

A seguir será mostrada a metodologia utilizada pela métrica de avaliação Carmen.

4.2.1.2.1 Análise fonética

Os conceitos de metrificação e rima referem-se a características fonéticas das palavras organizadas em versos e estrofes no poema, como visto em seção 2.1. São estes recursos estilísticos que imprimem no texto características sonoras, ritmo e musicalidade, inerentes a este estilo.

Contudo, existe uma inconsistência natural entre a grafia das palavras e sua fonética, *i.e.*, a pronúncia das palavras nem sempre pode ser inferida a partir de sua escrita. Por exemplo, apesar de terem a grafia similar, o som do grafema “nexo” nas palavras “inexorável” e “anexo” são distintas. Essa inconsistência também ocorre, de maneira até mais acentuada, em outros idiomas, como o inglês. Por exemplo, o grafema “oo” em “*boot*”, “*book*”, “*blood*”, “*brooch*” e “*door*”. Desta forma, torna-se desafiador realizar a avaliação de características fonéticas com base na representação gráfica das palavras.

A solução encontrada para contornar este obstáculo foi fazer um mapeamento entre as palavras e suas pronúncias, utilizando um dicionário de pronúncias para tanto. Infelizmente, não foi encontrado um dicionário de pronúncias em português adequado ao propósito, o que motivou a utilização da língua inglesa como base para a construção da métrica de avaliação, deixando a ressalva de que basta utilizar um dicionário de pronúncias em português para poder utilizá-la neste idioma. A criação de um dicionário fonético em português será abordado na seção 5.1: Trabalhos Futuros.

A partir disto, optou-se então por utilizar o *Carnegie Mellon University Pronouncing Dictionary* (CMUDict)⁸. Este dicionário traz a representação fonética de 134.000 palavras em inglês norte-americano. Cada palavra tem sua representação fonética mapeada para o alfabeto fonético ARPAbet, reconhecido como um padrão de pronúncia em inglês. A representação das vogais possui, além do som, a ênfase ou entonação que cada uma tem na pronúncia, podendo variar entre 0:

⁸ Disponível em: <<http://www.speech.cs.cmu.edu/cgi-bin/cmudict>>. Acesso em: 11.jun.2024.

sem ênfase, 1: ênfase principal e 2: ênfase secundária. No Quadro 20 são ilustradas algumas representações fonéticas de palavras.

Quadro 20 – Exemplos de representações fonéticas de palavras segundo o dicionário CMUdict.

Palavra	Representação Fonética
<i>love</i>	L - AH1 - V
<i>houston</i>	HH - Y - UW1 - S - T - AH0 - N
<i>honor</i>	AA1 - N - ER0
<i>forward</i>	F - A01 - R - W - ER0 - D
<i>computer</i>	K - AH0 - M - P - Y - UW1 - T - ER0
<i>science</i>	S - AY1 - AH0 - N - S
<i>architecture</i>	AA1 - R - K - AH0 - T - EH2 - K - CH - ER0

Fonte: (POSSI *et al.*, 2023)

Usando este mapeamento, torna-se possível analisar características fonéticas das palavras, permitindo a continuidade do desenvolvimento da métrica de avaliação.

4.2.1.2.2 Análise de rima

Como visto em seção 2.1, a rima é um recurso estilístico capaz de conferir musicalidade ao poema. Embora haja várias formas de sua ocorrência no texto, a mais comum é chamada rima externa, *i.e.*, quando ocorre entre as palavras finais dos versos (OLIVEIRA H., 2017). Desta a forma, nesse primeiro instante a métrica de avaliação CARMEN foca somente em rimas externas, analisando se as últimas palavras dos versos possuem sons semelhantes.

Esta análise é feita da seguinte forma: dado um poema com n linhas (versos), a métrica de avaliação analisa o esquema rímico comparando as palavras finais de cada verso com as $n - 1$ outras. Duas palavras são consideradas rimas quando suas m -últimas sílabas fonéticas coincidem, de acordo com sua representação fonética extraída do dicionário. O valor m representa o nível de semelhança entre as rimas e é um parâmetro da métrica de avaliação.

Quadro 21 – Exemplos de avaliação de rima entre duas palavras segundo a metodologia da métrica CARMEN.

Palavra A	Palavra B	Rima?	
		$m=1$	$m=2$
<i>science</i>	<i>appliance</i>	sim	não
<i>computer</i>	<i>persecutor</i>	sim	sim
<i>show</i>	<i>now</i>	não	não

Fonte: (POSSI *et al.*, 2023)

A métrica de avaliação então identifica todas as palavras finais dos versos que rimam, agrupando-as pelo fonema e identificando-as com letras maiúsculas

em ordem alfabética, extraíndo assim o esquema rímico do poema. A letra “X” é reservada para identificar aqueles versos que não rimam com nenhum outro.

Uma vez que o esquema rímico R_i tenha sido identificado, a avaliação do quesito rima A_{rhyme} poderá ocorrer de duas maneiras, dependendo da modalidade da métrica CARMEN escolhida: caso seja a CARMEN-d, existirá um esquema rímico esperado R_e e a avaliação, no quesito rima, será dada pelo número de versos nos quais a rima identificada é a rima esperada V_e dividido pelo total de versos no poema V_t , como ilustrado na Equação 4.1.

$$A_{rhyme} = \frac{V_e}{V_t} \quad (4.1)$$

Caso a modalidade utilizada seja a CARMEN-i, a avaliação da rima é feita subtraindo-se a unidade pelo número de versos que não rimam com nenhum outro V_x , dividido pelo número de versos que rimam com algum outro V_r , como ilustrado em Equação 4.2.

$$A_{rhyme} = 1 - \frac{V_x}{V_r} \quad (4.2)$$

Nas duas modalidades, o valor de A_{rhyme} irá variar de 0 (pior avaliação) à 1 (melhor avaliação).

4.2.1.2.3 Análise da metrificação do verso

A metrificação em um poema diz respeito às características das palavras que irão dar ritmo à leitura do texto, como mostrado em seção 2.1. Dentre estas características está a contagem de sílabas poéticas, a cadência e a cesura. Em que se pese a importâncias destas últimas características, o foco da métrica de avaliação será a primeira por questão de simplificação do problema. As outras características serão apontadas na seção 5.1: Trabalhos Futuros.

Embora a escansão seja feita contando-se as sílabas fonéticas, optou-se por contar as sílabas ortográficas até a última sílaba tônica do verso, por questão de eficiência e por considerar que esta é uma aproximação adequada do número de sílabas poéticas.

O método para metrificação dos versos é o seguinte: cada verso v_i do poema tem suas palavras convertidas em suas representações fonéticas e, a partir disto, são contadas as sílabas, as quais são representadas com números pelo dicionário fonético, obtendo-se o número de sílabas de cada verso s_i . A partir daí, este número é comparado com o número de sílabas esperado em cada verso se_i ,

conforme Equação 4.3. No caso da modalidade CARMEN-d, esse número é informado *a priori* para cada verso.

$$A_{metric} = 1 - \frac{\sum_{i=1}^{V_t} |s_i - se_i|}{\sum_{i=1}^{V_t} se_i} \quad (4.3)$$

No caso da modalidade CARMEN-i, o valor de se é calculado como a moda dos valores s_i previamente calculados. Caso a moda seja representada por mais de um elemento, é escolhido aquele mais próximo da média dos valores de s_i , conforme Equação 4.4.

$$A_{metric} = 1 - \frac{\sum_{i=1}^{V_t} |s_i - Mo|}{\sum_{i=1}^{V_t} Mo} \quad (4.4)$$

onde Mo é a moda dos valores de s_i .

Assim como na avaliação da rima, o valor A_{metric} irá variar de 0 (pior avaliação) à 1 (melhor avaliação).

4.2.1.2.4 Avaliação do poema

Após a obtenção dos valores de A_{rhyme} e A_{metric} , a pontuação total do poema é dada pela média aritmética destes, conforme Equação 4.5.

$$A = \frac{A_{rhyme} + A_{metric}}{2} \quad (4.5)$$

Foi utilizada a média aritmética inicialmente considerando que ambas as avaliações, tanto da metrificação dos versos quanto das rimas, têm o mesmo peso na avaliação do poema. Contudo, é possível dar pesos diferentes para cada um destes critérios.

O Quadro 22 ilustra a avaliação de um poema pela métrica. Foi utilizado a modalidade CARMEN-i que inferiu que os versos eram decassílabos ($Mo = 10$) e o esquema rímico ABABCDCDEFEBB. A avaliação do quesito rima foi de 100% ($A_{rhyme} = 1,0$) e do quesito metrificação foi 93% ($A_{metric} = 0,93$). No total, a métrica avaliou o poema com a pontuação de 96,5%.

Quadro 22 – Exemplo de avaliação de um poema pela métrica.

Verso	Metrificação	Esquema Rímico
<i>Thy gift, thy tables, are within my brain</i>	10	A
<i>Full characterized with lasting memory,</i>	07	B
<i>Which shall above that idle rank remain,</i>	10	A
<i>Beyond all date, even to eternity:</i>	11	B
<i>Or, at the least, so long as brain and heart</i>	10	C
<i>Have faculty by nature to subsist;</i>	10	D
<i>Till each to razed oblivion yield his part</i>	11	C
<i>Of thee, thy record never can be missed.</i>	10	D
<i>That poor retention could not so much hold,</i>	10	E
<i>Nor need I tallies thy dear love to score;</i>	10	F
<i>Therefore to give them from me was I bold,</i>	10	E
<i>To trust those tables that receive thee more:</i>	10	F
<i>To keep an adjunct to remember thee</i>	10	B
<i>Were to import forgetfulness in me.</i>	06	B

Trata-se do soneto 122 de William Shakespeare (1564-1616). Foi utilizado a modalidade CARMEN-i. A métrica identificou que os versos eram decassílabos e inferiu o esquema rímico.

Fonte: (POSSI *et al.*, 2023)

4.2.2 Construção do *corpus* de poemas: Carmen Corpus

Dentre os critérios que se acredita serem fundamentais para um *corpus* ser considerado adequado para o treinamento de modelos de linguagem, podemos destacar dois: a quantidade de amostras e a qualidade destas amostras em representar o que se quer modelar (MANNING; SCHUTZE, 1999). Um *corpus* robusto, com inúmeros poemas, é fundamental para fornecer aos modelos de linguagem a amplitude de dados necessária para aprender padrões linguísticos complexos e nuances da linguagem poética. Com esse critério, garante-se que os modelos sejam expostos a diversas variantes de estilo, estrutura e temática, aprimorando sua capacidade de gerar poemas originais e de alta qualidade. A qualidade, por sua vez, dos poemas presentes no *corpus* é igualmente importante, por influenciar diretamente o desempenho dos modelos de linguagem em abstrair as informações.

Contudo, essas propriedades são antagônicas. Para se obter um *corpus* com um tamanho considerado relevante para os modelos de linguagem atuais, usualmente lança-se mão de métodos automáticos de aquisição de textos. Contudo, ao aumentar a quantidade indiscriminadamente, arrisca-se diminuir a qualidade geral do *corpus*.

Uma das formas de contornar este problema é fazendo uma pré-seleção nos poemas candidatos a participarem do *corpus*; outra, que não exclui a primeira abordagem, é anotar os poemas com informações relevantes ao modelo.

Contudo, seria contraproducente, moroso, dispendioso e suscetível a erros avaliar ou anotar manualmente um *corpus* que foi obtido de forma automática.

Desta forma, optamos por uma abordagem automática para aquisição, seleção e anotação, admitindo apenas aqueles poemas bem avaliados em uma métrica específica e, posteriormente, anotando-os, ganhando eficiência e capacidade de lidar com grandes volumes de dados. Essa abordagem permite garantir a qualidade do *corpus*, a dimensão do mesmo e a representatividade das informações, possibilitando assim a construção de um *corpus* otimizado para o treinamento de modelos de linguagem.

A construção do Carmen Corpus partiu dos poemas obtidos do *corpus* Poems Dataset (NLP) (subseção 4.1.1.3). A escolha deste *corpus* como ponto de partida se deu por ser composto por poemas individualizados, com textos mais limpos, sem anotações ou marcações de qualquer espécie, e na maioria das vezes, sem prefácios, título ou nome do autor, o que permitiu utilizar o *corpus as-is*, e realizar a seleção dos poemas individualmente. Apesar de ser um *corpus* de menor porte, se comparado a *corpora* como o do Projeto Gutenberg (subseção 4.1.1.1), espera-se que resulte em um *corpus* mais adequado, após a etapa de seleção, do que os outros *corpora* candidatos, já que apresenta uma qualidade maior em suas amostras, em sua maior parte, já em formato de versos.

4.2.2.1 Seleção de poemas

Como mencionado na subseção 4.2.1, para a avaliação dos poemas candidatos a participarem do novo *corpus* foi utilizada uma métrica específica para esta forma de texto, a CARMEN (POSSI *et al.*, 2023), descrita na subseção 4.2.1.2. Esta métrica é capaz de fazer a avaliação automática do poema (utilizando sua variante CARMEN-i), além de inferir o esquema de rima utilizado e a métrica de cada verso, o que será útil mais tarde na anotação do *corpus*.

Todos os poemas presentes no *corpus* original passaram por uma etapa de avaliação e posterior seleção a partir da nota obtida na avaliação pela CARMEN. Apenas os poemas que obtiveram uma avaliação maior ou igual a 70% foram admitidos no novo *corpus*. Essa pontuação de corte foi definida empiricamente.

O Quadro 23 ilustra um poema retirado do *corpus* Poems Dataset (NLP) após avaliação realizada pela métrica CARMEN. O poema mostrado recebeu a avaliação 93% e, portanto, será admitido no novo *corpus*. O Gráfico 2, por sua vez, ilustra o perfil das avaliações dos poemas presentes do *corpus* Poems Dataset (NLP). Os poemas que obtiveram uma avaliação superior a 70% formam o grupo dos poemas aprovados.

Quadro 23 – Exemplo de um poema do *corpus* Poems Dataset (NLP) sendo avaliado pela métrica Carmen (POSSI *et al.*, 2023).

A shepherdess with a lamb to her breast
 Against a dry stone wall on a windy hill
 She cradles it with love so truly blessed
 Over steep Welsh hills they roam best
 Hardy hill sheep bred to wander at will
 A shepherdess with a lamb to her breast
 Long fleeces waving in springs unrest
 Bleating lambs at teats taking their fill
 She cradles it with love so truly blessed
 A brooding landscape the ultimate test
 The valley below a patchwork sits still
 A shepherdess with a lamb to her breast
 Her love is her flock she tenders with zest
 Caring for her lambs with love and skill
 She cradles it with love so truly blessed
 A shepherdess her lambs so caressed
 Her life with her sheep is always a thrill
 A shepherdess with a lamb to her breast
 She cradles it with love so truly blessed

Esquema de rima: ABAABAABAABAABAA
 Pontuação rima: 1,00
 Metrificação: 10
 Pontuação metrificação: 0,86
 Pontuação do poema: 0,93

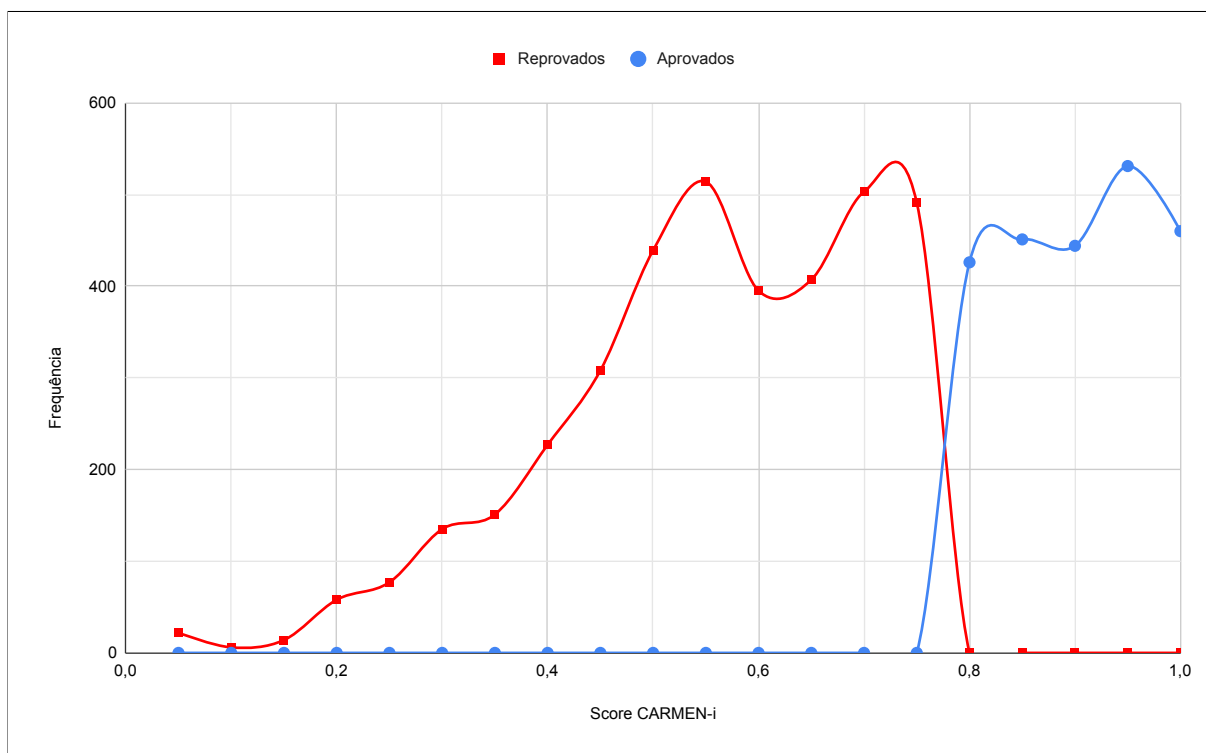
4.2.2.2 Anotação do *corpus*

Existem basicamente duas formas de se adicionar metadados ao *corpus*: adicionar os dados ao próprio *corpus*, por exemplo, prefixando ou pós-fixando o texto de forma que o modelo aprenda a relevância de tais dados durante o seu treinamento, ou alterar a arquitetura da rede neural para especializá-la de tal forma que os metadados passem a fazer parte da sua estrutura (LAU *et al.*, 2018b; GHAZVININEJAD *et al.*, 2016), seja com adição de mais camadas de abstração, seja com a adição de neurônios (parâmetros) específicos para cada metadado, por exemplo. Dado o objetivo de propor um *corpus* com anotações, optou-se pela primeira opção.

A primeira etapa para anotar o *corpus* foi decidir quais metadados deveriam ser acrescentados. Em se tratando de um *corpus* de poemas, dois metadados se destacam como imprescindíveis: a rima e a metrificação de cada verso.

Para a rima, optou-se por adicionar, em cada verso, a qual “grupo rimante” (grupo de palavras com a mesma sonoridade) do esquema de rimas o verso pertence, de forma a sinalizar para o modelo, mesmo que indiretamente, que a terminação daquele verso está relacionada com o grupo. Já a metrificação é informada de forma numérica também para cada verso, tendo em vista que alguns poemas podem ter versos com metrificações distintas entre si. Outro motivo para a escolha de se adicionar as anotações para cada verso é garantir que os metadados estejam sempre presentes na janela de atenção do modelo.

Gráfico 2 – Perfil dos poemas selecionados.



No gráfico é apresentado a comparação entre a pontuação dos poemas aprovados e os reprovados. Os poemas aprovados foram aqueles que obtiveram uma pontuação maior ou igual à 70% na métrica CARMEN-i. Do total de 6.295 poemas, 40,48% foram aprovados pela métrica, passando a constituir o novo *corpus*. A avaliação média no novo *corpus* aumentou 6,34% em relação ao *corpus* original.

Por fim, os experimentos mostraram evidências de que a rede tinha dificuldades em assimilar a informação de que cada poema era um elemento distinto dos demais. Para contornar este problema, cada poema passou a receber um identificador numérico, único por poema. Ao final dos mesmos, foi adicionado o *token* `<|endoftext|>`, que indica o final de uma sequência textual para alguns modelos, como o GPT-2.

No cabeçalho (primeira linha) de cada poema foi adicionado o formato do poema, sua metrificação e o esquema de rimas completo, com o intuito de facilitar o “*prompt*” da forma do poema desejado, além de disponibilizar ao modelo mais algumas características.

Todos esses metadados foram adicionados de forma prefixada aos versos, separados pelo caractere “|” (*pipe*). A escolha deste caractere se deu pelo mesmo ser raramente identificado nos textos e pela facilidade da sua utilização. Esse formato de anotação também facilita a utilização do *corpus* por pesquisadores que queiram eliminar os metadados ou utilizar apenas alguns. Fazê-los prefixados também auxilia aos modelos, que possuem janelas de contexto limitadas, a não

Quadro 24 – Exemplos de poemas marcados no *corpus*.

quatrain 6 ABAB	
4700 1 06 A	You may think its a rose
4700 2 05 B	But that would be wrong
4700 3 06 A	Is there someone who knows
4700 4 05 B	It wont take too long
< endoftext >	
sonnet 10 ABABCDCDEFEGG	
5398 1 10 A	Lo as a careful housewife runs to catch
5398 2 08 B	One of her featherd creatures broke away
5398 3 10 A	Sets down her babe and makes an swift dispatch
5398 4 10 B	In pursuit of the thing she would have stay
5398 5 10 C	Whilst her neglected child holds her in chase
5398 6 10 D	Cries to catch her whose busy care is bent
5398 7 10 C	To follow that which flies before her face
5398 8 08 D	Not prizing her poor infants discontent
5398 9 09 E	So runnst thou after that which flies from thee
5398 10 10 F	Whilst I thy babe chase thee afar behind
5398 11 10 E	But if thou catch thy hope turn back to me
5398 12 10 F	And play the mothers part kiss me be kind
5398 13 09 G	So will I pray that thou mayst have thy Will
5398 14 10 G	If thou turn back and my loud crying still
< endoftext >	
triolet 8 ABAAABAB	
5989 1 08 A	Let the sun shine brightly each day
5989 2 09 B	Warming us deeply in our soul
5989 3 08 A	We are happy its here to stay
5989 4 08 A	Let the sun shine brightly each day
5989 5 09 A	Giving our earth a lovely ray
5989 6 08 B	Having found its own final knoll
5989 7 08 A	Let the sun shine brightly each day
5989 8 09 B	Warming us deeply in our soul
< endoftext >	

O quadro ilustra três poemas retirados do novo *corpus* com suas anotações. Na primeira linha de cada poema é possível observar o cabeçalho do mesmo, que traz a forma, a metrificação dos versos e o esquema de rima do poema. Em seguida vêm os versos, precedidos pela anotação que contém o identificador único do poema, o número da linha do verso, a metrificação daquele verso e qual o grupo rimante do mesmo. Por fim, na última linha, a palavra-chave que identifica o final do poema.

“esquecerem” os metadados, já que, por estarem anotados em cada verso, provavelmente sempre estarão na janela de contexto. O Quadro 24 ilustra como é o resultado a marcação automática de um poema.

Esse processo foi repetido para todos os elementos do *corpus* original e o resultado foi um novo *corpus* aprimorado e com anotações. A avaliação média do novo *corpus* na métrica CARMEN foi de 80%, um aumento de 6,34% ($p < 0,001$) em relação à avaliação do *corpus* original.

4.2.3 Resultados

Nesta seção serão mostrados os experimentos realizados com o intuito de evidenciar o ganho de desempenho obtido utilizando o novo *corpus* no treina-

mento (*fine-tuning*) de modelos de linguagem.

Para os experimentos foi utilizado o modelo de linguagem pré-treinado GPT-2 (RADFORD *et al.*, 2019), baseado na arquitetura Transformer. Para o processamento foi utilizado uma instância do Google Colab com GPU A100. Cada modelo utilizado nos experimentos foi treinado por cerca de 12 horas.

4.2.3.1 Análise qualitativa do modelo

No primeiro experimento, duas instâncias da GPT-2 foram treinadas, uma com o *corpus* aprimorado, onde foram admitidos apenas os poemas que obtiveram avaliação maior ou igual a 70% e com a adição de anotações, e a outra foi treinada com o *corpus* original Poem Dataset (NLP), que fora utilizado como base, contendo todos os poemas e sem anotações.

Nos poemas ilustrados no Quadro 25 é possível notar evidências de que o modelo de linguagem conseguiu assimilar informações sobre forma, metrifcação e rima. Verifica-se que o texto foi separado em versos, que cada verso foi iniciado um número identificador único para o poema, um identificador único para o verso, a metrifcação de 10 sílabas para cada verso e o esquema de rima conforme solicitado no *prompt* do modelo, *i.e.*, o modelo está gerando versos também com metadados.

Como discutido na seção (subseção 2.2.3.2) de requisitos para geração de poemas, está fora do escopo do trabalho a avaliação do conteúdo poético dos poemas gerados. Contudo, é interessante notar também a presença consistente de figuras de linguagem, como metáforas e metonímias.

Contudo, como discutido na seção 3.1, a codificação da arquitetura utilizada para testes não expôs ao modelo as informações fonéticas das palavras, motivo pelo qual não se verifica rimas consistentes com o esquema enumerado. Embora possam ser verificadas algumas rimas, são provavelmente oriundas de memorização do *corpus* e não criadas pelo modelo.

Em todo caso, comparando esses poemas com os gerados pela instância treinada com o *corpus* original (Quadro 26), é notável que as alterações no *corpus* aprimorado fizeram com que o modelo tivesse um desempenho superior na tarefa de geração de poemas, notadamente pelo fato de que o segundo modelo mostrou uma tendência em gerar textos em prosa, ainda que gerasse quebras-de-linha aleatoriamente.

Quadro 25 – Poemas gerados com o *corpus* aprimorado.

Me: write a poem about the night.		sonnet 10 ABABCD	
2928 1 10 A	Poetry is for boys and girls play		
2928 2 10 B	Time the playmate of sense and truth		
2928 3 10 A	When the sky sings softly for days away		
2928 4 10 B	Lovely lines drawn on a blank canvas		
2928 5 10 C	A rhyme of words with no need for ornament		
2928 6 10 D	All day the sun and moon dance side by side		
2928 7 10 C	In a ballad of the heartbeats sweet harmony		
2928 8 10 D	Lovely lines drawn on a blank canvas		
Me: write a poem about love.		sonnet 10 ABABDCDEE	
3478 1 10 A	I' m not sure what words to use if I' m alone		
3478 2 08 B	Then I' ll put my pen to paper and give it a try		
3478 3 10 A	In answer to this letter I' d put pen to paper		
3478 4 10 B	And Im going to try writing poems again		
3478 5 10 C	Or maybe if I' m tired of poems I' d just make a rhyme		
3478 6 10 D	I hope you will give my poetry a try and leave me some solace		
3478 7 10 C	Theres just so much you can do in life and in writing		
Me: write a poem about love.		sonnet 10 ABABDCDEE	
3204 1 10 A	How can we love that which is most true		
3204 2 09 B	as if our own hearts somehow hold it in		
3204 3 09 A	and we would surely do so again		
3204 4 09 B	I hope it does not go astray		
3204 5 07 C	we just should not forget to share		
3204 6 09 D	the words with each other everyday		
Me: write a poem about aging.		sonnet 10 ABABCD	
4199 1 10 A	A lot of things I hate are now out of sight		
4199 2 10 B	So to write a poem about age is to die		
4199 3 10 A	But I dont want to live that long unless I die		
4199 4 10 B	Like the summer sun but with a beard		
4199 5 10 C	I want a poem of my own with age		
4199 6 10 D	My poems to tell the story of years old		
4199 7 10 C	I want to paint a picture of things that are new		
4199 8 10 D	I want to live my life in an old testament		

Neste quadro são ilustrados quatro poemas gerados com o modelo treinado com o Carmen Corpus, para serem comparados com os poemas gerados com o modelo treinado pelo *corpus* original, o Poems Dataset, ilustrado no Quadro 26. Na instrução ao modelo também foi adicionado o cabeçalho contendo o formato, a metrficação dos versos e o esquema de rima desejados. Todas as demais linhas, inclusive as marcações dos versos, foram gerados pelo modelo. Interessante notar que as marcações geradas pelo modelo têm um identificador único que se repete, a numeração sequencial dos versos, a metrficação igual à solicitada e o grupo rimante correto para cada verso, embora a rima não esteja consistente com o esquema, provavelmente devido às limitações discutidas em seção 3.1.

Fonte: (POSSI *et al.*, 2024?)

Quadro 26 – Poemas gerados com o *corpus* original.

<p>Me: write a poem about love.</p> <p>it' s not a sentimental one. The love is sincere, sincere, sincere and something I' ve wanted to write about for years. So it' s fun to write about it, because you can' t just slam it on the wall. And that' s fine, because the best way to get my words across is to write a song that rhymes about love, not just about love itself. No, it' s a poem that is meant to be sung. It' s a love story, not an allegory: I feel as if it' s written to capture the essence of a woman that is willing and willing to give her best meeting, and that' s why I continue to write about it.</p>
<p>Me: write a poem about love.</p> <p>My mom used to say, 'Write poems.' My mom also used to say, 'Write poems.' My dad always said, 'Write poems.' I got a BA in Public Affairs from Harvard and worked as an activist and writer. My mother also taught my friends and me how to write. My brother works at The Norton Foundation for Writers and Artists. My sister studied at University of California, where she earned a BA in English. My brother taught English at The University of Southern California. My dad taught at The Norton Foundation for Writers and Artists. My brother also works at The Norton Foundation for Writers and Artists.</p>
<p>Me: write a poem about aging.</p> <p>Rhyme scheme: ababcc; ababcd; abbacc; aabbcc The aabbcc will be Heroic Reza syllabic Verse. A simple refrain of 5th line may be as 6th line; if no refrain, the 6th line will be an individual one. Copyright © Muzahidul Reza 28 May, 2018 Tiras koi mujhko, tumne se hoon bhoolte Kabhi kisi sakte aati to zindagi yahan Mujhe dil chahti jaun kheyalay. Main kho gaye si kaht nikalay, Udhte chahti jaun kheyalay.</p>

Neste quadro são ilustrados três poemas gerados com o modelo treinado com o *corpus* Poems Dataset, para serem comparados com os poemas gerados com o modelo treinado com o Carmen Corpus, ilustrado no Quadro 25. Nele é possível notar que os poemas gerados pelo *corpus* original mostram uma tendência em gerar prosa em versos. O último exemplo, em especial, mostra que o modelo começou a gerar textos no idioma Hindi.

4.2.3.2 Análise quantitativa do modelo

O objetivo do segundo experimento foi avaliar quantitativamente o modelo treinado a partir do novo *corpus*, comparando-o com modelos treinados com os três *corpora* relacionados na subseção 4.1.1.

Para tanto, foram treinados quatro modelos de linguagem, um para cada *corpus*. A partir destes, foram geradas 45 amostras de poemas sobre 9 temas considerados recorrentes na literatura poética. Em seguida, essas amostras foram avaliadas pela métrica CARMEN. Os resultados para cada tema podem ser verificados na Tabela 2, e as médias das avaliações na Tabela 1.

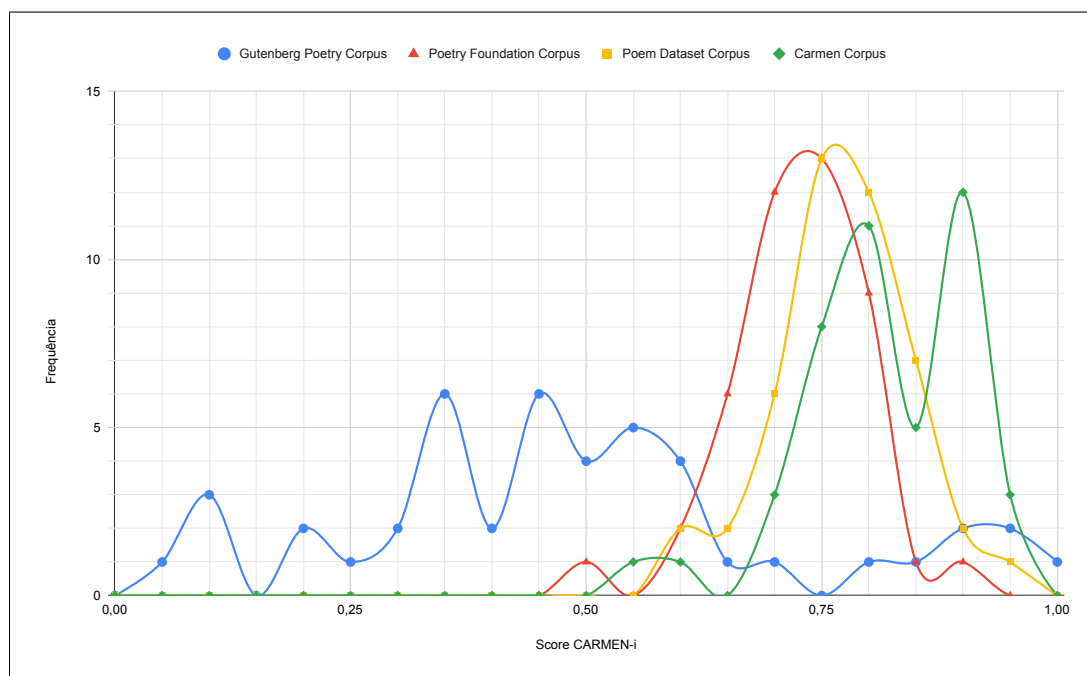
Os resultados evidenciam que as amostras geradas pelo modelo treinado

Tabela 1 – Avaliação média dos modelos pela métrica CARMEN-i.

Modelo	Avaliação Média			Comparação	
	Métrica	Rima	Score CARMEN-i	Ganho	p
Gutenberg	0,48	0,46	0,47	70,42%	<0,001
Poetry Foundation	0,53	0,87	0,70	13,59%	<0,001
Poems Dataset	0,67	0,83	0,75	6,34%	0,007
Carmen Corpus	0,87	0,73	0,80	-	-

Na tabela são mostradas as pontuações médias segundo a métrica CARMEN-i na avaliação das amostras de poemas geradas pelos modelos de linguagem treinados em cada *corpus* estudado. A coluna “ganho” representa o percentual de melhoria alcançado pela avaliação do modelo que utilizou o Carmen Corpus em relação aos demais, e seu respectivo valor- p , segundo o Teste T de Welch.

com o Carmen Corpus superou em até 70% (Tabela 1) as amostras geradas pelos outros modelos. No Gráfico 3 são ilustrados os perfis de cada conjunto de amostras: nele é possível notar que as amostras geradas pelo modelo treinado com o Carmen Corpus tem um grande viés à direita, evidenciando uma maior frequência de amostras com pontuações mais próximas do máximo.

Gráfico 3 – Comparação do desempenho dos modelos na métrica CARMEN-i.

No gráfico é apresentado a comparação entre os desempenhos dos modelos segundo a pontuação obtida pela métrica CARMEN-i (Tabela 2). A escala de pontuação, variando entre 0 e 1, é mostrada no eixo horizontal. A frequência (número de amostras) com que cada modelo atingiu uma determinada escala é mostrada no eixo vertical.

Tabela 2 – Avaliação dos modelos pela métrica CARMEN-i.

Tema	Avaliação dos <i>Corpora</i>							
	Gutenberg		Poetry Foundation		Poems Dataset		Carmen Corpus	
	Métrica	Rima	Métrica	Rima	Métrica	Rima	Métrica	Rima
beleza	0,44	0,52	0,27	0,65	0,75	0,84	0,89	0,84
	0,12	0,74	0,84	0,95	0,71	0,77	0,92	0,56
	0,52	0,00	0,52	0,92	0,73	0,82	0,87	0,73
	0,82	0,98	0,38	0,85	0,79	0,95	0,94	0,78
	0,86	0,95	0,50	0,89	0,76	0,92	0,69	0,83
morte	0,56	0,57	0,41	0,77	0,65	0,93	0,84	0,86
	0,90	0,94	0,46	0,77	0,60	0,76	0,90	0,58
	0,07	0,61	0,66	0,91	0,73	0,61	0,91	0,89
	0,59	0,43	0,63	0,94	0,66	0,90	0,90	0,70
	0,47	0,82	0,56	0,81	0,77	0,73	0,85	0,65
espiritualidade	0,37	0,33	0,64	0,90	0,73	0,77	0,89	0,51
	0,24	0,86	0,60	0,81	0,71	0,61	0,86	0,59
	0,69	0,25	0,61	0,80	0,74	0,77	0,93	0,25
	0,50	0,20	0,55	0,78	0,81	0,76	0,91	0,70
	0,51	0,62	0,66	0,80	0,57	0,66	0,85	0,89
esperança	0,36	0,27	0,66	0,86	0,59	0,84	0,82	0,71
	0,68	0,68	0,57	0,85	0,48	0,91	0,85	0,68
	0,59	0,50	0,61	0,93	0,61	0,82	0,87	0,89
	0,18	0,00	0,45	0,95	0,80	0,86	0,87	0,63
	0,68	0,14	0,46	0,91	0,72	0,87	0,87	0,93
identidade	1,00	0,00	0,48	0,85	0,93	0,97	0,86	0,20
	0,66	0,00	0,40	0,93	0,68	0,74	0,90	0,89
	0,03	0,13	0,53	0,87	0,66	0,98	0,87	0,50
	0,86	0,00	0,66	0,96	0,74	0,82	0,86	0,79
	0,26	0,47	0,58	0,92	0,50	0,61	0,83	1,00
amor	0,37	0,50	0,49	0,95	0,46	0,77	0,87	0,56
	0,21	0,18	0,65	0,89	0,64	0,95	0,86	0,90
	0,40	0,33	0,42	0,82	0,57	0,85	0,78	0,76
	0,36	0,81	0,59	0,76	0,80	0,69	0,84	0,72
	0,99	0,98	0,63	0,91	0,68	0,96	0,91	0,70
natureza	0,72	0,25	0,48	0,81	0,61	0,77	0,87	0,93
	0,30	0,53	0,56	0,94	0,69	0,93	0,89	0,58
	0,60	0,45	0,46	0,86	0,53	0,89	0,92	0,82
	0,03	0,50	0,60	0,92	0,58	0,82	0,82	0,72
	0,26	0,56	0,38	0,91	0,70	0,80	0,87	0,67
sofrimento	0,11	0,39	0,57	0,84	0,39	0,77	0,80	0,65
	0,15	0,00	0,46	0,90	0,78	0,79	0,82	0,57
	0,72	0,95	0,32	0,95	0,71	0,79	0,87	0,97
	0,05	0,00	0,59	0,90	0,81	0,89	0,86	0,73
	0,47	0,62	0,62	0,86	0,73	0,92	0,92	0,86
tempo	0,77	0,94	0,61	0,82	0,71	0,91	0,89	0,67
	0,38	0,00	0,57	0,94	0,53	0,90	0,84	0,90
	0,69	0,46	0,50	0,90	0,65	0,90	0,88	0,95
	0,42	0,20	0,23	0,91	0,64	0,78	0,86	0,78
	0,68	0,83	0,54	0,88	0,66	0,93	0,88	0,78

Na tabela são mostradas as pontuações segundo a métrica CARMEN-i, tanto no quesito métrica quanto no quesito rima, obtidas pelas amostras de poemas geradas pelos modelos de linguagem treinados em cada *corpus* estudado. Foram escolhidos 9 temas recorrentes em poemas e, para cada tema, foram geradas 5 amostras.

4.2.4 Discussão

Os resultados evidenciam que as amostras geradas pelo modelo treinado com o Carmen Corpus superou em até 70% (Tabela 1) as amostras geradas pelos outros modelos.

Uma questão que nos chama atenção nos experimentos é o fato da média no quesito rima ter sido maior nas amostras geradas pelo modelo treinado com

o *corpus* Poems Dataset. Contudo, examinando as amostras, é possível verificar que isto se deve ao fato de haver muitas frases iguais sendo repetidas continuamente em alguns poemas. Neste cenário, a métrica de avaliação irá analisar se os versos estão rimando e têm a mesma metrifcação de verso, pontuando-os por isso. Seria interessante adotar algum método para penalizar estes casos, como evitar contabilizar rimas entre palavras idênticas. Essa característica será abordada na seção 5.1: Trabalhos Futuros. Além disso, como discutido na seção 3.1, não era esperado um ganho expressivo no quesito rima pelo fato dos experimentos terem sido realizados com um modelo que utiliza o *encoding* BPE.

Em todo caso, os resultados foram positivos, mostrando que mesmo no cenário atual de LLMs de propósito geral pré-treinados em *datasets* massivos, ainda há espaço para *corpora* elaborados para problemas específicos e que estes podem aumentar ainda mais os desempenhos dos grandes modelos atuais.

4.3 CarmenGen, terceira versão

Em face das dificuldades encontradas nas versões anteriores, em particular no que diz respeito a gerar rimas, decidiu-se por propor uma terceira versão do modelo, cujo objetivo é equilibrar a geração de conteúdo poético através da previsão de *tokens* com base na distribuição probabilística aprendida pela rede a partir de um *corpus* de textos selecionados enquanto aplica restrições para garantir que o texto gerado atenda aos critérios especificados. Dessa forma, o modelo conseguiria gerar textos poéticos com conteúdo significativo relacionado ao *prompt* do usuário enquanto adere aos padrões definidos por meio de hiperparâmetros do modelo, eliminando a necessidade de um modelo separado para cada combinação possível de forma de poema, esquema de rima ou metrifcação.

A construção do modelo foi então dividida em duas partes: a criação e treinamento do modelo-base a partir do *corpus* com textos selecionados, de forma que o modelo aprenda a gerar textos com certa perplexidade (subjetivos, poéticos), e a implementação do algoritmo para geração condicional de texto com base nos parâmetros especificados.

4.3.1 Desenvolvimento do modelo-base

Para a criação do modelo, decidiu-se por utilizar o modelo de linguagem GPT-2 (RADFORD *et al.*, 2020) como ponto de partida, reutilizando assim seu treinamento inicial (*transfer-learning*) e, em seguida, realizar o *fine-tuning* do modelo em um *corpus* de poemas selecionados.

Este modelo foi escolhido devido ao seu bom desempenho na geração de

texto e sua capacidade de ser executado em *hardware* mais modesto. Enquanto modelos maiores como ChatGPT-3.5 ou 4.0 oferecem recursos e desempenho avançados, GPT-2 pode ser uma escolha mais prática para *fine-tuning* em cenários onde eficiência computacional, custo e restrições de recursos são fatores críticos. Além disso, o GPT-2 permite estender sua funcionalidade para implementar a geração condicional mais facilmente.

O modelo de linguagem GPT-2 evidenciou, em trabalhos anteriores, seu desempenho em diversas tarefas, alcançando o estado da arte em 7 das 8 em que foi testado (RADFORD *et al.*, 2019). Embora a proposta do modelo seja ser multitarefa, pronto para usar e o mais genérico possível, esta grande generalidade fez com que o modelo, paradoxalmente, tenha limitações quando se é necessário gerar textos que cumpram certos requisitos mais específicos, como definir quais palavras devem obrigatoriamente aparecer (ou quais devem ser omitidas), qual o nível de simplicidade das palavras escolhidas (e conseqüentemente das frases geradas), o tamanho médio das sentenças *etc.* Muitos autores tentam alcançar esses objetivos realizando o *fine-tuning* em *corpora* específicos, muitas das vezes lançando mão de artifícios para criar estruturas artificiais nestes *corpora* objetivando criar uma tendência na rede em gerar os textos com os critérios desejados.

Esse método, de utilizar-se de interferências propositais no *corpus* a fim de criar estruturas artificiais, embora seja muito utilizado e gere alguns bons resultados, além de não assegurar que o texto será gerado com os critérios desejados, impede que sejam criadas ferramentas de uso geral que poderiam ser treinadas (ou especializadas, através do *fine-tuning*) com *datasets* brutos, como visto na subseção 2.2.2.2.1.

Para se ter maior controle sobre o processo de geração do texto, mantendo-se todo o poderio do modelo GPT-2, propõe-se uma abordagem supervisionada, utilizando uma busca heurística, *i.e.*, a partir da distribuição probabilística aprendida pela rede neural, um algoritmo percorra os diversos caminhos possíveis, buscando as palavras com maior probabilidade, como o modelo faria, mas considerando também outros critérios especificados.

4.3.1.1 Desenvolvimento do *corpus*

Para treinar o modelo, decidiu-se por utilizar um *corpus* composto exclusivamente de poemas com metrificação e rima, visando treinar o modelo para gerar textos em versos. Como visto nas versões anteriores, isso por si só não é suficiente para gerar poemas com esses atributos. No entanto, essa abordagem reduz a probabilidade de o modelo tentar gerar prosa em versos, e aumenta a sua probabilidade de gerar um conteúdo poético, com a presença de figuras de

linguagem e outros recursos estilísticos deste tipo de conteúdo.

O *corpus* utilizado foi similar ao Carmen Corpus, gerado na subseção 4.2.2, com exceção das anotações que, nesse momento, não foram utilizadas pelo modelo.

4.3.1.2 Treinamento do modelo

O modelo GPT-2 foi treinado (*fine-tuning*) usando a versão do Carmen Corpus sem anotações (subseção 4.2.2). O modelo foi treinado por 10 épocas. O Quadro 27 ilustra um exemplo de um poema gerado pelo modelo livremente (sem geração condicional) após o treinamento.

Quadro 27 – Exemplo de poema gerado pelo modelo treinado com o *corpus* de poemas selecionados.

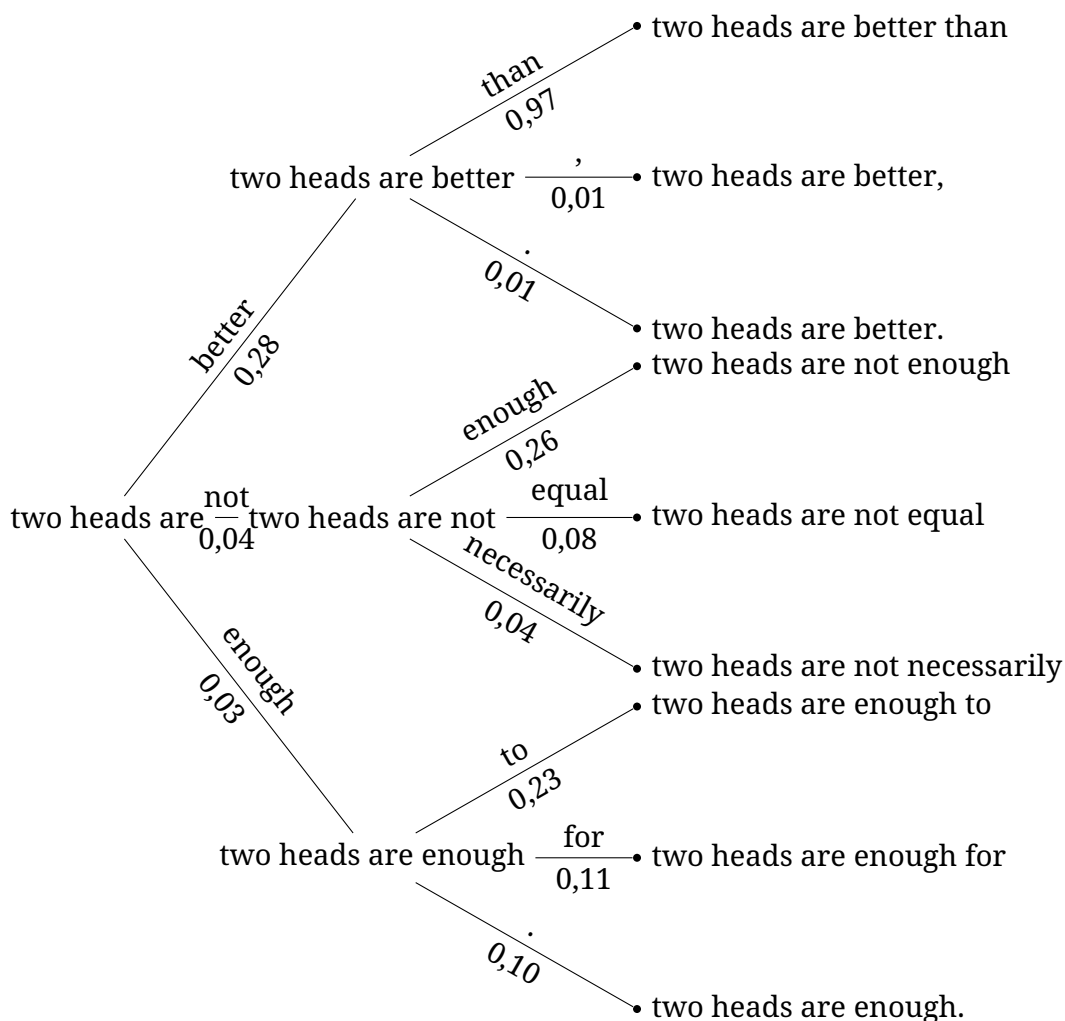
```
There was a country orchard near Poitiers Gate
Where two small children played a song
Accosting one who unexpectedly
Besetteting the birds and setting about
Strolling on a milkwhite mare so pale and fair
He whistled the green for every bird that flew
And never a bird replied with a laugh
Excuse the pun somewhat I will spare you the pain
Of having to find out every legend old
Yet its a matter of tedious rereading
For a while it was only a plain storybook
But now when I look back over my shoulder
And see the things that made poetry rich
```

4.3.2 Modelo de geração condicional de texto

O GPT-2 permite acesso às pontuações de predição (*prediction scores*) através da classe interna auxiliar `GPT2LMHeadModel`. A partir daí, a cada passo é possível obter todas as possibilidades de próximas palavras a partir da sentença gerada até o momento, possibilitando reconstruir o grafo que representa as cadeias de Markov com todas as possibilidades e suas probabilidades a partir da posição atual. A Figura 2 mostra um exemplo deste tipo de grafo para a entrada “*two heads are*”: a rede geraria provavelmente a sentença “*two heads are better than*”, supondo uma geração determinística que considera sempre a escolha das palavras de maior probabilidade, contudo, a partir do grafo, outras opções poderiam ser exploradas no intuito de se atingir algum outro critério, como escolher palavras específicas, impedir que algumas palavras sejam escolhidas, ou, no caso em estudo, escolher palavras que formem um verso com metrifcação e rima.

A partir disso, a geração do texto torna-se um problema de busca heurística de caminho em grafos, já bem estabelecido na Ciência da Computação, cujo objetivo será maximizar a função heurística gerando, em consequência, sentenças que atendam a um determinado critério. No estudo de caso proposto, isto é, a

Figura 2 – Próximas palavras sugeridas pelo GPT-2 e suas probabilidades.



A imagem ilustra um grafo representando as previsões do modelo GPT-2 (próximas palavras e suas probabilidades) a partir da sentença inicial “two heads are”.

geração de poemas, o algoritmo terá como objetivo gerar sentenças que tenham o número de sílabas igual à metrificação definida e que apresente a rima segundo o padrão escolhido, levando também em consideração a distribuição estatística aprendida pela rede neural, o que deverá contribuir para a gramaticalidade e os valores semântico e poético dos versos.

4.3.2.1 Modelando o problema

Para simplificar o problema, definiu-se que a granularidade da solução seria gerar um verso de cada vez que atendesse aos critérios definidos. Assim, resolver o problema de gerar um poema com quatro versos será equivalente a resolver quatro problemas de gerar um verso conforme as especificações. Também definiu-se que a rima ocorreria sempre em sua forma mais comum e identificável, no final dos versos, e se usaria uma aproximação da metrificação fonética

silábica com base em sílabas ortográficas.

Os hiper-parâmetros do modelo para gerar um verso são: o contexto inicial, o número de sílabas que ele deve ter e qual deve ser sua rima.

O contexto inicial diz respeito aos versos gerados anteriormente e ao *prompt* com as instruções para o modelo. Essas instruções podem influenciar o conteúdo semântico do texto, introduzindo, por exemplo, um tema ou emoção específica.

O número de sílabas se refere ao comprimento do verso. Embora os poemas geralmente tenham versos do mesmo comprimento, há estilos como *haiku* e *limerick*, onde há uma restrição de comprimento diferente para determinados versos dentro do mesmo poema. Como esse parâmetro é especificado para cada verso, o modelo também é capaz de gerar esses tipos de poemas.

As informações de rima também são especificadas para cada verso e podem ser “livre”, o que significa que o verso pode ter qualquer final, ou pode indicar outro verso com o qual o atual deve rimar. Por exemplo, se o intuito é gerar um poema com quatro versos e esquema rímico ABAB, as restrições de rima seriam: verso 1, livre; verso 2, livre; verso 3: rimar com verso 1; verso 4, rimar com verso 3.

4.3.2.2 Análise fonética

Tanto o comprimento dos versos quanto a rima das palavras dependem da análise fonética das palavras. No entanto, muitas técnicas de incorporação não consideram a fonética das palavras, como o caso do BPE usado no GPT-2, conforme discutido na seção 3.1.

Para superar essa limitação, utilizou-se a mesma abordagem descrita na criação da métrica CARMEN (subseção 4.2.1.2), que consiste em incorporar um dicionário fonético ao modelo, o CMUDict (CMU, 2023).

4.3.2.3 Algoritmo de geração condicional

Uma vez modelado o problema, definiu-se o algoritmo proposto para resolvê-lo. A proposta é baseada no algoritmo *best-first search*. O comportamento do algoritmo está sinteticamente ilustrado no Algoritmo 1.

Com base nos parâmetros iniciais, o algoritmo fornece ao modelo base (GPT-2) o contexto (C) e o conteúdo atual (v_a) para obter a previsão das próximas palavras (P) já ordenadas por suas probabilidades. A partir daí, o algoritmo inicia uma iteração sobre as palavras, começando pela com maior probabilidade. Na primeira etapa da iteração, ele adiciona a palavra de maior probabilidade (p) ao verso que está sendo gerado (v). Em seguida, ele avalia o verso atual para veri-

ficar se ele atende aos critérios especificados (como comprimento, rima *etc.*). Essa avaliação pode resultar em um sinal de interrupção para o algoritmo, que pode indicar quando o verso não será mais capaz de atender aos requisitos e o procedimento deve ser interrompido (ABORT) ou quando o verso atende aos requisitos e o procedimento pode ser concluído (GOAL). Caso contrário, se a avaliação indicar que o verso ainda não atingiu o objetivo, mas que ainda é viável, o procedimento continua recursivamente, chamando a função para continuar gerando o verso, com a palavra escolhida nesta iteração anexada a ele, e com a metrificação alvo (M) reduzida pelo comprimento da palavra recém-adicionada ao verso.

Algoritmo 1: Algoritmo para decodificação condicional de um verso.

```

1 Function Verse( $C, M, R, v_a$ )
2    $P \leftarrow \text{ModelGenerate}(C, v_a)$ ;
3   for  $p \in P$  do
4      $v \leftarrow v_a + p$ ;
5      $a \leftarrow \text{Evaluate}(C, M, R, v)$ ;
6     if  $a = [\text{ABORT}]$  then
7       | continue;
8     end
9     if  $a = [\text{GOAL}]$  then
10    | return  $v$ ;
11    end
12     $m \leftarrow \text{Measure}(p)$ ;
13    return Verse( $C, M - m, R, v$ );
14  end
15  return [ABORT];
16 end

```

Fonte: (POSSI *et al.*, 2024a)

Este é o funcionamento básico do algoritmo. Sua implementação em Python é baseada no algoritmo *beam search* para explorar múltiplos caminhos simultaneamente, aplica os conceitos de *sampling* com *temperature* e *top-k*, e considera a probabilidade acumulada dos caminhos, além de outros refinamentos apresentados a seguir.

4.3.2.4 Previsão de próximas palavras

A previsão das próximas palavras e suas probabilidades é feita pelo modelo base, GPT-2, que foi ajustado no *corpus* de poemas selecionados. O texto gerado até o momento é passado para o modelo para gerar as próximas palavras e suas probabilidades. As probabilidades das palavras são calculadas usando a função Softmax (Equação (2.7)) a partir dos *predicting scores* informados pelo modelo.

Em seguida, a lista de próximas palavras é filtrada para remover previsões de símbolos gráficos, números e fragmentos de palavras, deixando apenas previsões de palavras completas e de alguns sinais de pontuação. Outros *tokens* específicos do modelo também são removidos. Em seguida, palavras cuja probabilidade está abaixo do limite mínimo p_{min} são removidas. Finalmente, apenas as n principais palavras com as maiores probabilidades são selecionadas. Tanto p_{min} quanto n são hiper-parâmetros do modelo. Algumas palavras também são removidas aleatoriamente (com base em suas probabilidades) da lista de palavras selecionadas, o que provou-se empiricamente evitar degenerações no modelo.

4.3.2.5 Avaliação do verso

Ao incorporar cada nova palavra, o verso é submetido à avaliação da sua metrificação e da sua rima. A avaliação inicial verifica o comprimento do verso c_v : se esse comprimento exceder o comprimento solicitado M (considerando um parâmetro de tolerância t_m), o verso é descartado. Se, no entanto, esse comprimento estiver inserido na faixa crítica, *i.e.*, se $M - t_m \leq c_v \leq M + t_m$, indicando que o verso está se aproximando do fim, então a rima do verso é avaliada.

A avaliação da rima dependerá do tipo de critério especificado para o verso atual. Se o critério for “rima livre”, a avaliação da rima consistirá em descartar o verso se a última palavra não tiver rimas suficientes. A quantidade mínima de rimas que uma palavra precisa ter para ser a última palavra de um verso é um hiper-parâmetro do modelo. Essa otimização evita a geração de versos que seriam impossíveis de rimar no futuro. Se o critério for que o verso precisa rimar com outro, então é verificado se a última palavra do verso que está sendo avaliado rima com a última palavra do verso com o qual ele deve rimar, respeitando a tolerância especificada também como hiper-parâmetro do modelo, que indica o nível necessário de similaridade de fonemas para considerar que duas palavras rimem.

4.3.3 Resultados

Para avaliar o modelo quantitativamente, foram geradas vinte amostras de poemas⁹ que atendiam aos seguintes critérios: doze poemas com quatro versos decassilábicos e um esquema de rima ABAB, e oito poemas com versos dodecassilábicos e o mesmo esquema de rima ABAB. O índice de sucesso do modelo em gerar os poemas conforme os critérios especificados foi de 95%, sinalizando quanto a impossibilidade de encontrar o poema nos casos de insucesso. Os poe-

mas gerados foram então avaliados usando a métrica específica de avaliação de poemas CARMEN (POSSI *et al.*, 2023). Os resultados são apresentados na Tabela 3. O Quadro 28 ilustra algumas amostras dos poemas gerados pelo modelo.

Tabela 3 – Avaliação de poemas gerados pelo modelo utilizando a métrica CARMEN.

Amostra	Avaliação CARMEN-i		
	Métrica	Rima	Total
1	0,97	0,50	0,74
2	0,90	1,00	0,95
3	0,93	1,00	0,97
4	0,74	0,50	0,62
5	1,00	1,00	1,00
6	0,96	1,00	0,98
7	0,96	0,50	0,73
8	0,92	1,00	0,96
9	0,96	0,50	0,73
10	0,92	0,50	0,71
11	1,00	1,00	1,00
12	0,96	1,00	0,98
13	1,00	1,00	1,00
14	0,97	1,00	0,99
15	0,95	1,00	0,98
16	0,95	1,00	0,98
17	0,97	1,00	0,99
18	0,92	1,00	0,96
19	1,00	1,00	1,00
20	0,97	1,00	0,99
Média	0,95	0,88	0,91

Analisando os resultados, é possível notar que as amostras receberam uma avaliação média de 91%, com a avaliação parcial média para rima sendo 88% e a avaliação parcial média para metrificação sendo 95%.

É importante notar que a avaliação dos resultados do modelo apresentada aqui é superior à avaliação apresentada no trabalho Possi *et al.* (2024a). Isso se deu devido a uma correção no procedimento de avaliação de rimas da métrica CARMEN (POSSI *et al.*, 2023), que não considerava como rima palavras como “*chain*” e “*again*”. A correção foi pontual e não altera a metodologia da métrica descrita no artigo.

Analisando textualmente as amostras (Quadro 28), percebe-se que os poemas apresentam conteúdo poético, muitas vezes exibindo figuras de linguagem como metáforas, metonímias e analogias. Além disso, os textos são dispostos em versos, com comprimentos mais uniformes e rimas no padrão solicitado. Essas características são destacadas pela pontuação obtida na métrica CARMEN-i (Tabela 3), onde o modelo obteve uma avaliação média de 91%. As pontuações mais

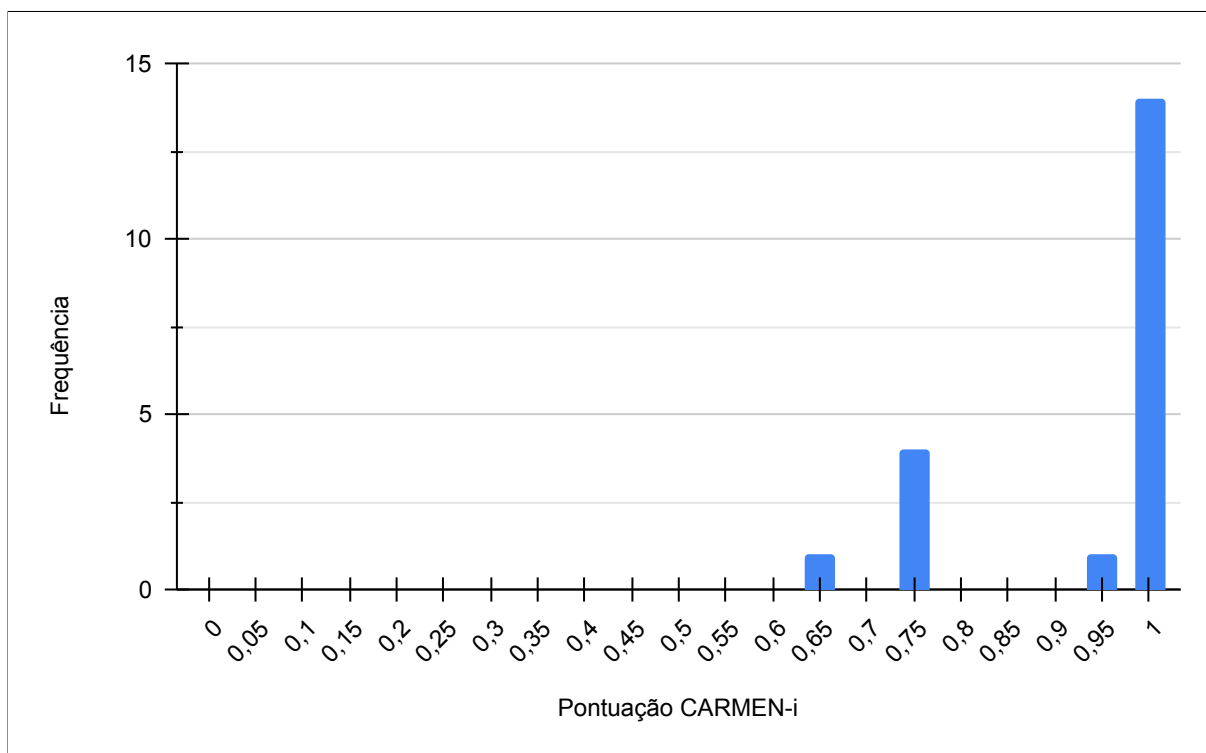
⁹ Esse foi o número de amostras possíveis no espaço de tempo disponível para os testes.

Quadro 28 – Amostras de poemas geradas pelo modelo

A poem is like a chain symbol to the whole of life and the poem is a refrain symbol to the entire mind	My heart beats faster than I could ever be free and I feel like a burden to all of the other three
There are a thousand ways to find happiness in the light of a few rays of your love that shine even	I can only pray to God he forgive my soul of sin and bring me into the new light and be whole
There is something more to come in the coming days but for now I am too far gone to know the ways	My life is not a dream game of chance or a game game to win or a team game of chance or to blame
There are a lot of different kinds of people in America today who do not know what it is that defines what they do or want to do in everyday	
There is something that I cannot explain beyond the simple words that come to the mind when we think of love and pain and all the ways in which we see too	
The clock ticks a certain hour and thirty minutes before it ticks back in time for you to get to the end and start the day right here in this black	
Your heart beat faster with every beat it beats faster with every kiss it lets go and I can feel the heat on your cheeks and you are soaked with bliss	
There is something more to life than just sitting around and enjoying life as if it were the end in itself most certainly not the end of all strife	
I can only pray for you all the best that we do go on our way and that we all do live a life worthy of those we have lost in the turmoil away	

São apresentadas seis amostras de poemas com quatro versos decassílabos e sete amostras de poemas com quatro versos dodecassílabos. Todos com padrão rímico ABAB e sobre diferentes temas. Os fonemas rimantes estão destacados em negrito.

frequentes foram 100% e 75% (Gráfico 4), indicando a capacidade do modelo em gerar consistentemente poemas no formato requerido.

Gráfico 4 – Histograma de pontuações obtidas pelas amostras.

Histograma ilustrando a distribuição de frequência de pontuações das amostras obtidas através da métrica CARMEN-i. As pontuações mais frequentes foram 100% e 75%.

Fonte: Possi *et al.* (2024b)

4.3.4 Discussão

Os experimentos mostraram que o modelo é capaz de gerar poemas consistentemente, alcançando uma média de 91% em uma métrica específica para avaliação de poemas. Adicionalmente, a análise textual evidenciou que os poemas, além dessas características formais, apresentam conteúdo poético no qual é possível encontrar figuras de linguagem e um estilo linguístico típico da poesia.

Aliado a isso, o modelo gerou poemas com rimas e versos metrificados de forma consistente, superando uma das principais limitações dos trabalhos relacionados. O trabalho de Popescu-Belis *et al.* (2023), por exemplo, gerava rimas no padrão AABB em 60% das vezes, e no padrão ABAB em apenas 45% das vezes. O modelo proposto gera a rima no padrão solicitado em 95% das vezes e, quando não consegue gerar, informa a impossibilidade ao invés de gerar o poema fora do padrão.

Por fim, pode-se destacar que o modelo não necessita de um *corpus* específico de poemas de um formato ou de um sentimento, sendo possível utilizá-lo para gerar qualquer formato e sobre qualquer tema, bastando para isso ajustar os hiperparâmetros do modelo, incluindo o *prompt* inicial. Essa característica

supera a limitação de trabalhos como o de Hopkins e Kiela (2017), que em sua primeira versão necessitava de um *corpus* para cada tipo de poema que se almejava gerar, e o de Bena e Kalita (2019), que tinha um modelo treinado para cada emoção.

Durante os testes, verificou-se que havia um erro na detecção de rimas da métrica CARMEN, que foi corrigido. Os testes desta versão foram reavaliados utilizando a versão corrigida da métrica. O erro não afetou as versões anteriores.

5 Considerações Finais

O presente trabalho visou contribuir em uma área desafiadora do Processamento de Linguagem Natural: geração automática de poemas. Visando determinar se é possível gerar poemas automaticamente que apresentem metrifcação e rima, o trabalho culminou no desenvolvimento do modelo de linguagem CarmenGen, um modelo que apresenta avanços e demonstra a capacidade de gerar poemas que atendem a critérios específicos de forma, com evidências de conteúdo poético. O modelo, baseado em redes neurais artificiais, se destaca por sua capacidade de gerar poemas com metrifcação e rima consistentemente.

Os resultados obtidos com o CarmenGen, avaliados pela métrica CARMEN, demonstram a sua capacidade de gerar poemas, atingindo uma avaliação média de 91% em uma métrica específica de avaliação, gerando poemas que atendiam os critérios especificados em 95% das vezes, atingindo assim os objetivos propostos.

As principais contribuições do trabalho são:

- Proposta do modelo CarmenGen: Este modelo utiliza um algoritmo de geração condicional de texto que permite a especificação da forma, metrifcação e esquema de rima desejados;
- Desenvolvimento da métrica CARMEN: Uma métrica específica para avaliação automática de poemas, capaz de analisar quantitativamente a metrifcação dos versos e o esquema rímico;
- Criação do Carmen Corpus: Um *corpus* de poemas aprimorado e anotado automaticamente, contendo informações sobre a forma, metrifcação e rima de cada poema, que pode ser utilizado para treinar modelos de linguagem a gerar poemas automaticamente.

Além disso, encontraram-se evidências de que textos agradáveis esteticamente são mais inteligíveis, abrindo caminho para pesquisas de modelos de linguagem que produzam textos que possam ser compreendidos com maior facilidade pelo leitor. Outra evidência importante encontrada é que, mesmo no cenário atual de LLMs pré-treinados em *datasets* massivos, há espaço para *corpora* elaborados para problemas específicos.

Esses resultados representam um avanço significativo na área de geração automática de textos, demonstrando o potencial da PLN para criar sistemas capa-

zes de gerar poemas com qualidade e consistência. As contribuições do trabalho, incluindo o modelo CarmenGen, a métrica CARMEN e o Carmen Corpus, pavimentam o caminho para futuras pesquisas e aplicações nesta área.

5.1 Trabalhos Futuros

- Criar um dicionário de pronúncias em português que possa ser utilizado programaticamente;
- Melhorar a métrica CARMEN para que possa avaliar características como cadência e cesura nos poemas, e penalizar a rima entre palavras idênticas;
- Modificar o modelo para utilizar o *corpus* com anotações;
- Aprimorar a utilização de pontuação pelo modelo.

5.2 Publicações

Os seguintes trabalhos foram publicados durante o desenvolvimento desta tese:

- MOREIRA, A.; OLIVEIRA, A.; POSSI, M. The intersection between linguistic theories and computational linguistics over time. *DELTA: Documentação de Estudos em Lingüística Teórica e Aplicada*, v. 38, 2022. ISSN 1678460X.
- RODRIGUES, M.; OLIVEIRA, A.; MOREIRA, A.; POSSI, M. Lyrics generation supported by pre-trained models. In: *The International FLAIRS Conference Proceedings*, v. 35, 5 2022. ISSN 23340762.
- POSSI, M. OLIVEIRA, A.; MOREIRA, A.; COSTA, M. Carmen: A method for automatic evaluation of poems. In: *Proceedings of the IEEE 5th International Conference on Natural Language Processing (ICNLP 2023)*. 2023. p. 244–247. ISBN 9798350302219.
- POSSI, M. OLIVEIRA, A.; MOREIRA, A.; COSTA, M. A Neural Network-based Language Model for Automatic Poem Generation. In: *Proceedings of the IEEE 20th International Conference on Intelligent Computer Communication and Processing Conference (ICCP 2024)*. 2024.

5.3 Aspectos Éticos e Ambientais

Esta tese não envolveu a participação de seres humanos. Desta forma, não foi necessário submetê-la à revisão pelo Comitê de Ética em Pesquisa da Universidade Federal de Viçosa.

Contudo, as questões éticas que envolvem os modelos de linguagem para geração de texto certamente se aplicam também à geração de poemas, em especial, a geração de conteúdo ofensivo, a reprodução de estereótipos preconceituosos ou a substituição da criatividade humana por máquinas. Embora estas sejam questões complexas para as quais ainda não há soluções definitivas, esse trabalho se preocupou com estes aspectos e tentou mitigá-los sempre que possível. Quanto à geração de conteúdo ofensivo ou preconceituoso, em todos os testes realizados não se verificou a presença de tais conteúdos, de forma que não se mostrou necessário criar filtros para removê-los. Acreditamos que isso deva-se ao fato dos *corpora* utilizados. Quanto à substituição da criatividade humana, o objetivo do trabalho é criar um modelo que possa auxiliar e estimular a expressão criativa dos seres humanos, nunca a substituir.

Quanto ao consumo energético e ao impacto ambiental, o presente trabalho utilizou a técnica de *transfer learning*, que permitiu a reutilização de um modelo de linguagem treinado em dados massivos com alto custo energético, dissolvendo assim seu custo. Além disso, utilizou, sempre que possível, energia de fonte solar para o desenvolvimento dos modelos localmente. Durante o período do trabalho, foram utilizados o total de 5,86 MWh oriundos de painéis fotovoltaicos. Além disso, priorizou-se a utilização de modelos capazes de serem utilizados em *hardware* mais modestos, com menor consumo energético.

Referências

BAHDANAU, D.; CHO, K. H.; BENGIO, Y. Neural machine translation by jointly learning to align and translate. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, p. 1–15, 2015.

BENA, B.; KALITA, J. Introducing aspects of creativity in automatic poetry generation. In: SHARMA, D. M.; BHATTACHARYA, P. (Ed.). *Proceedings of the 16th International Conference on Natural Language Processing*. International Institute of Information Technology, Hyderabad, India: NLP Association of India, 2019. p. 26–35. Disponível em: <<https://aclanthology.org/2019.icon-1.4>>.

CAMÕES, L. *Rimas*. [S.l.]: Biblioteca Geral da Universidade de Coimbra, 1598.

CAMÕES, L. D. *Os lusíadas*. [S.l.]: Magalhães & Moniz, 1910.

CELIKYLMAZ, A.; CLARK, E.; GAO, J. Evaluation of text generation: A survey. *arXiv*, 2020. ISSN 23318422.

CHAKRABARTY, T.; PADMAKUMAR, V.; HE, H. Help me write a poem: Instruction tuning as a vehicle for collaborative poetry writing. In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. [S.l.: s.n.], 2022. p. 6848–6863.

CHAKRABARTY, T.; PADMAKUMAR, V.; HE, H. *Help me write a poem*: Instruction tuning as a vehicle for collaborative poetry writing. In: GOLDBERG, Y.; KOZAREVA, Z.; ZHANG, Y. (Ed.). *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, 2022. p. 6848–6863. Disponível em: <<https://aclanthology.org/2022.emnlp-main.460/>>.

CMU. *The CMU Pronouncing Dictionary*. [S.l.]: Carnegie Mellon University, 2023. <<http://www.speech.cs.cmu.edu/cgi-bin/cmudict>>. Accessed: 2023-12-31.

COSTA, S. R. *Dicionário de gêneros textuais*. [S.l.]: Autêntica, 2018.

CRUYS, T. V. de. Automatic poetry generation from prosaic text. In: . Association for Computational Linguistics, 2020. p. 2471–2480. Disponível em: <<https://www.aclweb.org/anthology/2020.acl-main.223>>.

DAS, A.; GAMBÄCK, B. Poetic machine: Computational creativity for automatic poetry generation in bengali. In: *ICCC*. [S.l.: s.n.], 2014. p. 230–238.

ELAM, M. Poetry will not optimize; or, what is literature to ai? *American literature*, Duke University Press, v. 95, n. 2, p. 281–303, 2023.

FAN, A.; LEWIS, M.; DAUPHIN, Y. Hierarchical neural story generation. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2018. p. 889–898. Disponível em: <<http://aclweb.org/anthology/P18-1082>>.

- GHAZVININEJAD, M.; SHI, X.; CHOI, Y.; KNIGHT, K. Generating topical poetry. In: . Association for Computational Linguistics, 2016. p. 1183–1191. ISBN 9781945626258. Disponível em: <<http://aclweb.org/anthology/D16-1126>>.
- GOODFELLOW, I. J.; POUGET-ABADIE, J.; MIRZA, M.; XU, B.; WARDE-FARLEY, D.; OZAIR, S.; COURVILLE, A.; BENGIO, Y. Generative adversarial nets. *Advances in Neural Information Processing Systems*, v. 3, n. January, p. 2672–2680, 2014. ISSN 10495258.
- GORENZ, D.; SCHWARZ, N. *How funny is ChatGPT? A comparison of human- and A.I.-produced jokes*. 2024.
- HOLTZMAN, A.; BUYS, J.; DU, L.; FORBES, M.; CHOI, Y. The curious case of neural text degeneration. *CEUR Workshop Proceedings*, v. 2540, 2019. ISSN 16130073.
- HOPKINS, J.; KIELA, D. Automatically generating rhythmic verse with neural networks. *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, v. 1, p. 168–178, 2017.
- IQBAL, T.; QURESHI, S. The survey: Text generation models in deep learning. *Journal of King Saud University - Computer and Information Sciences*, The Authors, n. xxxx, p. 1–14, 2020. ISSN 22131248. Disponível em: <<https://doi.org/10.1016/j.jksuci.2020.04.001>>.
- JAKESCH, M.; HANCOCK, J. T.; NAAMAN, M. Human heuristics for ai-generated language are flawed. *Proceedings of the National Academy of Sciences*, v. 120, 3 2023. ISSN 0027-8424.
- JURAFSKY, D.; MARTIN, J. H. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*. 3rd. ed. [s.n.], 2024. Draft. Disponível em: <<https://web.stanford.edu/~jurafsky/slp3/>>.
- KÖBIS, N.; MOSSINK, L. D. Artificial intelligence versus Maya Angelou: Experimental evidence that people cannot differentiate AI-generated from human-written poetry. *Computers in Human Behavior*, v. 114, n. September 2020, 2021. ISSN 07475632.
- LAU, J. H.; COHN, T.; BALDWIN, T.; BROOKE, J.; HAMMOND, A. Deep-speare: A joint neural model of poetic language, meter and rhyme. *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, v. 1, p. 1948–1958, 2018. ISSN 2331-8422.
- LAU, J. H.; COHN, T.; BALDWIN, T.; BROOKE, J.; HAMMOND, A. Deep-speare: A joint neural model of poetic language, meter and rhyme. *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, v. 1, p. 1948–1958, 2018. ISSN 2331-8422.
- LIN, C.-Y.; OCH, F. J. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In: *Proceedings of the 42nd annual meeting of the association for computational linguistics (ACL-04)*. [S.l.: s.n.], 2004. p. 605–612.

LINARDAKI, C. Poetry at the first steps of artificial intelligence. *Humanist Studies & the Digital Age*, v. 7, n. 1, 2022.

LOLLER-ANDERSEN, M.; GAMBÄCK, B. Deep Learning-based Poetry Generation Given Visual Input. *Proceedings of the Ninth International Conference on Computational Creativity (ICCC'18)*, p. 240–247, 2018.

LUONG, M. T.; PHAM, H.; MANNING, C. D. Effective approaches to attention-based neural machine translation. *Conference Proceedings - EMNLP 2015: Conference on Empirical Methods in Natural Language Processing*, p. 1412–1421, 2015.

MALMI, E.; TAKALA, P.; TOIVONEN, H.; RAIKO, T.; GIONIS, A. Dopelearning: A computational approach to rap lyrics generation. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. [S.l.: s.n.], 2016. p. 195–204.

MANNING, C.; SCHUTZE, H. *Foundations of statistical natural language processing*. [S.l.]: MIT press, 1999.

MANURUNG, H. An evolutionary algorithm approach to poetry generation. University of Edinburgh. College of Science and Engineering. School of ..., 2004.

MISZTAL, J.; INDURKHYA, B. Poetry generation system with an emotional personality. In: *Proceeding of the Fifth International Conference on Computational Creativity*. [S.l.]: ICC, 2014. p. 72–81.

MOREIRA, A.; OLIVEIRA, A.; POSSI, M. The intersection between linguistic theories and computational linguistics over time. *DELTA: Documentação de Estudos em Lingüística Teórica e Aplicada*, v. 38, 2022. ISSN 1678-460X.

NIGHTINGALE, S. J.; FARID, H. Ai-synthesized faces are indistinguishable from real faces and more trustworthy. *Proceedings of the National Academy of Sciences*, v. 119, 2 2022. ISSN 0027-8424.

OBERMEIER, C.; MENNINGHAUS, W.; KOPPFELDS, M. von; RAETTIG, T.; SCHMIDT-KASSOW, M.; OTTERBEIN, S.; KOTZ, S. A. Aesthetic and emotional effects of meter and rhyme in poetry. *Frontiers in Psychology*, v. 4, 2013. ISSN 16641078.

OLIVEIRA, G. H. Automatic generation of poetry : an overview Automatic generation of poetry : an overview. *1st Seminar of Art, Music, Creativity and Artificial Intelligence*, n. June, p. 1–6, 2009.

OLIVEIRA H., H. G. A survey on intelligent poetry generation: Languages, features, techniques, reutilisation and evaluation. In: *Proceedings of the 10th International Conference on Natural Language Generation*. Association for Computational Linguistics, 2017. p. 11–20. Disponível em: <<http://aclweb.org/anthology/W17-3502>>.

OLIVEIRA H., H. G.; HERVÁS, R.; DÍAZ, A.; GERVÁS, P. Multilingual extension and evaluation of a poetry generator. *Natural Language Engineering*, Cambridge University Press, v. 23, p. 929–967, 11 2017. ISSN 1351-3249. Disponível em: <https://www.cambridge.org/core/product/identifier/S1351324917000171/type/journal_article>.

- OLIVEIRA S., S. *Análise de textos literários: poesia*. [S.l.]: InterSaberes, 2017.
- ORMAZABAL, A.; ARTETXE, M.; AGIRREZABAL, M.; SOROA, A.; AGIRRE, E. Poelm: A meter-and-rhyme-controllable language model for unsupervised poetry generation. In: *Findings of the Association for Computational Linguistics: EMNLP 2022*. [S.l.: s.n.], 2022. p. 3655–3670. ISBN 2205.12206v1.
- PAPINENI, K.; ROUKOS, S.; WARD, T.; ZHU, W.-J. Bleu: a method for automatic evaluation of machine translation. In: *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. [S.l.: s.n.], 2002. p. 311–318.
- PETROV, A.; MALFA, E. L.; TORR, P. H. S.; BIBI, A. Language model tokenizers introduce unfairness between languages. In: *NEURIPS 2023. Proceedings of 37th Conference on Neural Information Processing Systems*. [S.l.], 2023.
- POPESCU-BELIS, A.; ATRIO, À. R.; BERNATH, B.; BOISSON, E.; FERRARI, T.; THEIMER-LIENHARD, X.; VERNIKOS, G. Gpoet: a language model trained for rhyme generation on synthetic data. In: . Association for Computational Linguistics, 2023. p. 10–20. Disponível em: <<https://aclanthology.org/2023.latechclfl-1.2>>.
- POSSI, M.; OLIVEIRA, A.; MOREIRA, A.; COSTA, M. Carmen: A method for automatic evaluation of poems. In: *Proceedings of the 5th International Conference on Natural Language Processing, ICNLP 2023*. IEEE, 2023. p. 244–247. ISBN 9798350302219. Disponível em: <<https://ieeexplore.ieee.org/abstract/document/10236873>>.
- POSSI, M.; OLIVEIRA, A.; MOREIRA, A.; COSTA, M. Carmen corpus: um corpus aprimorado e anotado automaticamente para geração de poemas em modelos de linguagem baseados em redes neurais. *Abakós*, 2024? No prelo.
- POSSI, M.; OLIVEIRA, A.; MOREIRA, A.; COSTA, L. A neural network-based language model for automatic poem generation. In: *Proceedings of the IEEE 20th International Conference on Intelligent Computer Communication and Processing Conference (ICCP 2024)*. [S.l.: s.n.], 2024.
- POSSI, M.; OLIVEIRA, A.; MOREIRA, A.; COSTA, M. A neural network-based language model for automatic poem generation. In: *Proceedings of the 20th International Conference on Intelligent Computer Communication and Processing, ICCP 2024*. [S.l.]: IEEE, 2024. p. 50.
- POTASH, P.; ROMANOV, A.; RUMSHISKY, A. Ghostwriter: Using an lstm for automatic rap lyric generation. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. [S.l.: s.n.], 2015. p. 1919–1924.
- PROVILKOV, I.; EMELIANENKO, D.; VOITA, E. Bpe-dropout: Simple and effective subword regularization. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. [S.l.: s.n.], 2020. p. 1882–1892.
- RADFORD, A.; NARASIMHAN, K.; SALIMANS, T.; SUTSKEVER, I. Improving language understanding by generative pre-training. 2018.
- RADFORD, A.; WU, J.; CHILD, R.; LUAN, D.; AMODEI, D.; SUTSKEVER, I. *et al.* Language models are unsupervised multitask learners. *OpenAI blog*, v. 1, n. 8, p. 9, 2019.

- RADFORD, I. S. A.; WU, J.; CHILD, R.; LUAN, D.; AMODEI, D. [gpt-2] language models are unsupervised multitask learners. In: *OSDI'04: Sixth Symposium on Operating System Design and Implementation*. [S.l.: s.n.], 2020. v. 1, p. 137–150.
- RASHEL, F.; MANURUNG, R. Pemuisi: a constraint satisfaction-based generator of topical indonesian poetry. In: *ICCC*. [S.l.: s.n.], 2014. p. 82–90.
- RODRIGUES, M. A.; OLIVEIRA, A.; MOREIRA, A.; POSSI, M. Lyrics generation supported by pre-trained models. *The International FLAIRS Conference Proceedings*, v. 35, 5 2022. ISSN 2334-0762.
- ROSA, J. G. Grande sertão: veredas. [texto definitivo]. *Rio de Janeiro: José oympio*, 1958.
- SAEED, A.; ILIĆ, S.; ZANGERLE, E. Creative gans for generating poems, lyrics, and metaphors. 2019. Disponível em: <<http://arxiv.org/abs/1909.09534>>.
- SUN, Y.; YANG, C.-H.; LYU, Y.; LIN, R. From pigments to pixels: A comparison of human and ai painting. *Applied Sciences*, v. 12, p. 3724, 4 2022. ISSN 2076-3417.
- TIAN, Y.; PENG, N. Zero-shot sonnet generation with discourse-level planning and aesthetics features. In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. [S.l.: s.n.], 2022. p. 3587–3597.
- TORFI, A.; SHIRVANI, R. A.; KENESHLOO, Y.; TAVAF, N.; FOX, E. A. Natural language processing advancements by deep learning: A survey. *arXiv*, p. 1–21, 2020. ISSN 23318422.
- TURING, A. Computing machinery and intelligence. *Mind*, v. 59, n. 236, p. 433–460, 1950.
- VASWANI, A.; SHAZEER, N.; PARMAR, N.; USZKOREIT, J.; JONES, L.; GOMEZ, A. N.; KAISER, Ł.; POLOSUKHIN, I. Attention is all you need. In: *Advances in Neural Information Processing Systems*. [S.l.: s.n.], 2017. v. 2017-December. ISSN 10495258.
- WANG, J.; ZHANG, X.; ZHOU, Y.; SUH, C.; RUDIN, C. There once was a really bad poet, it was automated but you didn't know it. *Transactions of the Association for Computational Linguistics*, MIT Press Journals, The, v. 9, p. 605, 3 2021.
- WONG, M. T.; CHUN, A. H. W.; LI, Q.; CHEN, S.; XU, A. Automatic haiku generation using vsm. In: CITESEER. *WSEAS International Conference. Proceedings. Mathematics and Computers in Science and Engineering*. [S.l.], 2008. v. 7.
- YAN, R. I, poet: Automatic poetry composition through recurrent neural networks with iterative polishing schema. *IJCAI International Joint Conference on Artificial Intelligence*, v. 2016-Janua, p. 2238–2244, 2016. ISSN 10450823.
- YAN, R.; JIANG, H.; LAPATA, M.; LIN, S.-D.; LV, X.; LI, X. I, poet: automatic chinese poetry composition through a generative summarization framework under constrained optimization. In: AAAI PRESS. *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence*. [S.l.], 2013. p. 2197–2203.

YU, L.; ZHANG, W.; WANG, J.; YU, Y. SeqGAN: Sequence generative adversarial nets with policy gradient. *31st AAAI Conference on Artificial Intelligence, AAAI 2017*, p. 2852–2858, 2017.

ZHANG, X.; LAPATA, M. Chinese poetry generation with recurrent neural networks. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. [S.l.: s.n.], 2014. p. 670–680.