Lorenzo Puppi Vecchi

Transferência entre estilos de texto usando CycleGAN com espaço latente de estilo supervisionado

Curitiba - PR, Brasil
2022

Lorenzo Puppi Vecchi

Transferência entre estilos de texto usando CycleGAN com espaço latente de estilo supervisionado

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de Mestre em Informática.

Pontifícia Universidade Católica do Paraná
Programa de Pós-Graduação em Informática

Orientador: Emerson Cabrera Paraiso
Coorientadora: Eliane C. Francisco Maffezzolli

Curitiba - PR, Brasil
2022

Sumário

Lista de Figuras

Lista de Tabelas

Lista de abreviações

TST	Transferência de estilo de textos
BLUE	*Bilingual Evaluation Understudy*
TSN	Transferência de estilo neural
TAN	Tradução automática neural
GAN	Rede neural adversária generativa

Abstract

Text style transfer is a relevant task, contributing to theoretical and practical advancement in several areas, especially when working with non-parallel data. The concept behind non-parallel style transfer is to change a specific dimension of the sentence while retaining the overall context. Previous work used adversarial learning to perform such a task. Although it was not initially created to work with textual data, it proved to be very effective. Most of the previous work has focused on creating algorithms capable of transferring between binary styles, with limited generalisation capabilities and limited applications. This work proposes a method capable of working with multiple styles and improving content retention (BLEU) after a transfer. A CycleGan architecture was used along with the separation of latent spaces of style and content. The proposed method uses supervised learning from the style latent space. The results show that the method performed better than the previous methods in terms of the BLEU metric. Our model obtained 57.2% and 46.2% in the databases used for testing, surpassing the best previous works that reached 32.2% and 22.4% in the same databases. Such results suggest that the proposed method improves content retention in multi-style scenarios, while maintaining comparable accuracy in other test metrics in relation to the state of the art.

Keywords: Text style transfer, Supervised learning, Generative adversarial network.

Resumo

A transferência de estilos de texto é uma tarefa relevante, contribuindo para o avanço teórico e prático em diversas áreas, principalmente quando se trabalha com dados não paralelos. O conceito por trás da transferência de estilo não paralelo é alterar uma dimensão específica da frase, mantendo o contexto geral. Trabalhos anteriores usaram aprendizagem adversária para realizar tal tarefa. Embora não tenha sido criado inicialmente para trabalhar com dados textuais, mostrou-se bastante eficaz. A maior parte do trabalho anterior concentrou-se na criação de algoritmos capazes de transferir entre estilos binários, com capacidades de generalização e aplicações limitadas. Este trabalho propõe um método capaz de trabalhar com múltiplos estilos e melhorar a retenção de conteúdo (BLEU) após uma transferência. Foi utilizada uma arquitetura CycleGan juntamente com a separação dos espaços latentes de estilo e conteúdo. O método proposto utiliza aprendizagem supervisionada para o espaço latente de estilo. Os resultados mostram que o método obteve uma performance superior aos métodos anteriores no que diz a respeito a métrica BLEU. Nosso modelo obteve 57.2% e 46.2% nas bases de dados utilizadas para teste, superando os melhores trabalhos anteriores que alcançaram 32.2% e 22.4% nas mesmas bases de dados. Tais resultados sugerem que o método proposto melhora a retenção de conteúdo em cenários multi-estilo, mantendo uma acurácia comparável nas outras métricas de teste em relação ao estado da arte.

Palavras-chave: Transferência de estilo de texto, aprendizagem supervisionada, rede adversária generativa.

1 Introdução

Diversas são as propriedades dos textos que têm sido objeto de estudo entre pesquisadores de várias áreas, desde a inteligência computacional até a comunicação e marketing. Dentre as diversas nuances de pesquisa que podem ser derivadas da compreensão geral dos textos com diversas aplicações, um aspecto que tem atraído a atenção mais recentemente é a transferência de estilo de textos (TST). Esta área é considerada relativamente nova e foi produto das demandas latentes de duas áreas de pesquisa: transferência neural de estilo (GATYS; ECKER; BETHGE, 2016) e tradução de texto com a utilização de algoritmos de aprendizagem profunda (BAHDANAU; CHO; BENGIO, 2014).

Alguns trabalhos da literatura abordam o tema de TST como forma de inverter o sentimento de uma frase, fazendo com que uma frase negativa adquira um caráter positivo e vice-versa, como exposto pela Figura 1. Esta é uma forma simplificada do potencial do processo de TST, que pode extrapolar a transferência binária.

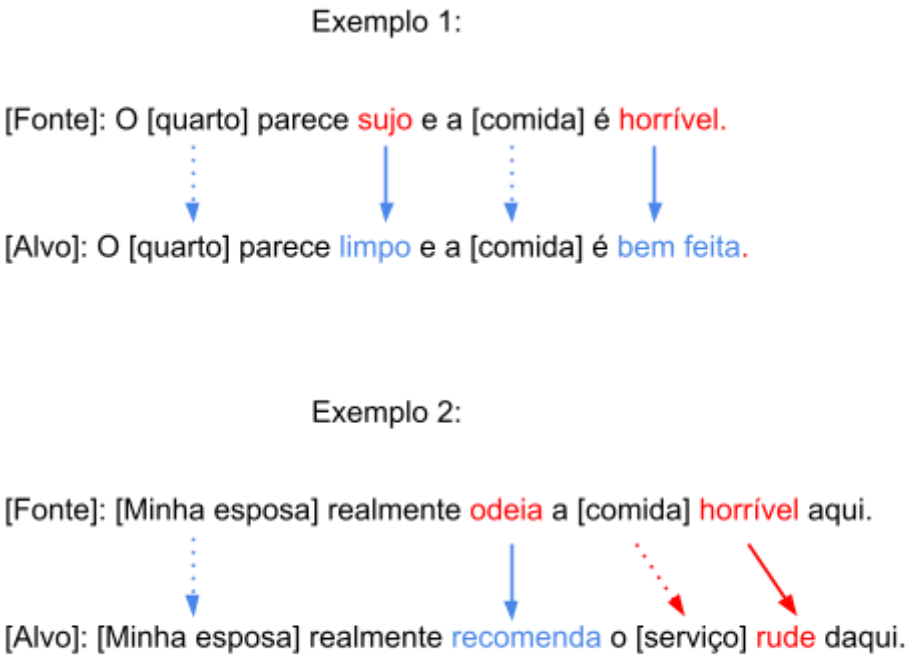

Figura 1. Exemplo de transferência de estilo (positivo e negativo).
(Extraído e adaptado de: (CHEN, J. *et al.*, 2019)).

A área e técnicas relacionadas a TST usualmente focam no desenvolvimento de algoritmos que têm a capacidade de alterar as propriedades estilísticas do texto,

mantendo seu estilo e conteúdo como variáveis independentes durante o processo. A habilidade de tornar dois aspectos de alto nível independentes durante o processo de criação de um algoritmo é um tópico igualmente relevante, visto que esta capacidade é uma tarefa complexa e, até então, era exclusiva dos humanos. O nome técnico que se dá a separação de dois aspectos de alto nível para a compreensão de um contexto mais amplo é “desentrelaçamento” ou “desemaranhamento dimensional” ou ainda “de características latentes” (CHEN, X. *et al.*, 2016). Este tópico está diretamente relacionado a transferência de estilo por conta da implícita diferenciação que deve ser apreendida entre conteúdo e estilo do texto para que a TST seja possível.

Na Figura 2 é possível perceber a utilização empírica do conceito de desentrelaçamento de características presentes em cada uma das imagens. Em certos momentos o algoritmo consegue alterar somente a cor do chão, em outro o formato do que originalmente era uma esfera, até a rotação da câmera pode ser alterada. Apesar de ser outra vertente de aplicação, com este exemplo simples fica mais explícita a aplicação e utilidade deste conceito. Muito além do exemplo na Figura 2, este conceito pode ser expandido para outras áreas e contexto de aplicação, como a TST, sendo o estilo e conteúdo as características latentes passíveis de serem mudadas independentemente.

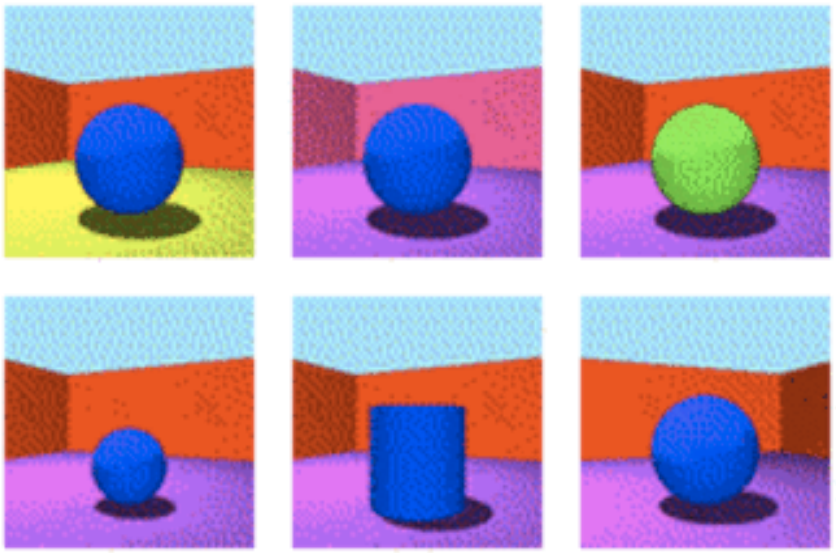

Figura 2. Exemplo de desentrelaçamento dimensional.
(Imagem extraída de: https://ai.googleblog.com/2019/04/evaluating-unsupervised-learning-of.html, em 30/06/2022).

Na Figura 3 pode-se perceber o processo genérico com uma aplicação mais representativa da TST. Nela podemos ver que o texto original é codificado em dois espaços diferentes, o estilo do texto e o conteúdo do texto. As dimensões latentes resultantes da codificação podem ser concatenadas de uma forma diferente para o passo de decodificação, como no exemplo da Figura 3, em que a dimensão de estilo é alterada para se gerar uma frase com uma polaridade de sentimento invertida.

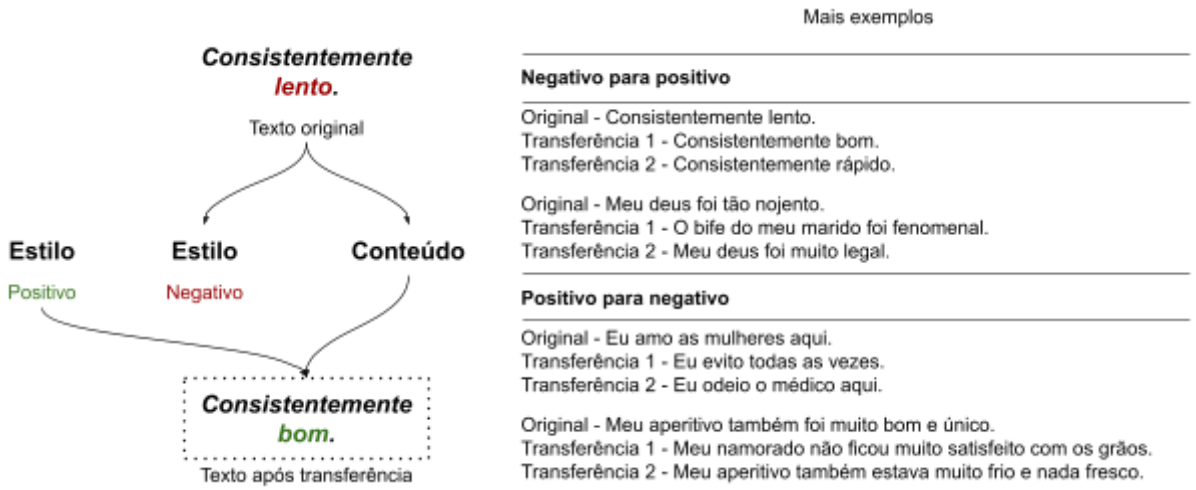

Figura 3. Exemplo do racional por trás da transferência de estilo.
(Quadro (direita) extraído de: (SHEN *et al.*, 2017)).

1.1 Motivação

Um problema que impacta diretamente nos algoritmos de transferência de estilo é a existência de corpus paralelos. Dados pareados são bases que possuem o suposto X e Y de forma estruturada para treinamento. No caso da TST isto representaria duas frases com exatamente o mesmo conteúdo, e somente o estilo modificado. Usualmente o que se encontra são múltiplas bases de dados, cada qual com seu estilo, porém sem o pareamento explícito. O termo *Non-parallel style transfer* trata justamente de algoritmos de TST com arquitetura de redes neurais capazes de serem treinadas nos ditos dados não paralelos (SHEN *et al.*, 2017). Reitera-se aqui a importância do desentrelaçamento das características de conteúdo e estilo.

Atualmente, a maioria dos métodos TST trabalham com bases não pareadas que se concentram na transferência do texto entre dois estilos. Acreditamos que os

estudos de TST devem ir além de realizar uma transferência de estilo binário e explorar tarefas mais ricas e dinâmicas. Por exemplo, (LAI *et al.*, 2019) propôs uma tarefa de transferência de estilo de atributos múltiplos onde um texto é transferido especificando atributos, como sentimento, gênero do autor, etc.

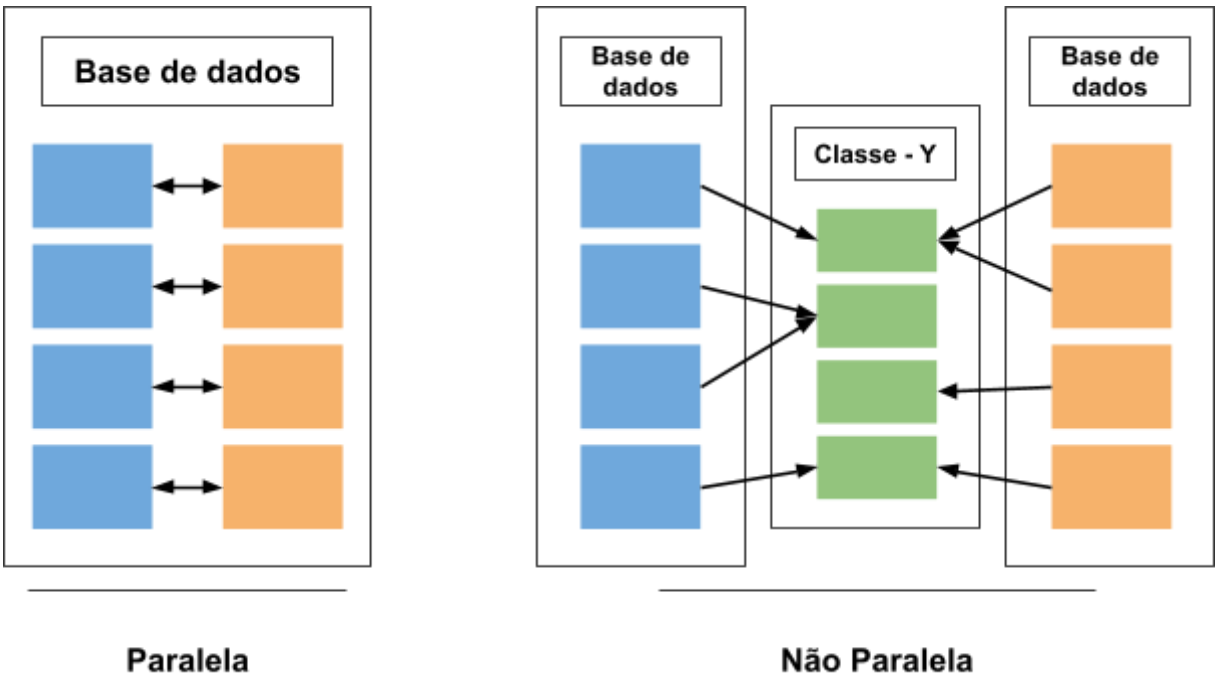

Figura 4.Bases de dados pareadas X não pareadas.
(Quadro (direita) extraído de: (BALTRUŠAITIS et al., 2019)).

Como visto na Figura 4, as bases de dados pareadas possuem exemplos para uma mesma instância quando esta se encontra em uma classe ou outra. Por exemplo se tivéssemos duas esferas e somente alterássemos a cor destas. Em uma base de dados não pareada, por sua vez, diferentes instâncias pertencem a uma mesma classe, porém não compartilham outras características. Um exemplo disso poderia ser a que uma esfera e um quadrado possuem a cor azul, mas são figuras diferentes. Este conceito pode ser traduzido para o campo da TST, visto que diferentes frases podem conter um mesmo estilo de redação ou característica, porém são semanticamente diferentes.

Para que seja possível medir a eficácia dos algoritmos de TST, assim como abordado por (HU, Zhiting et al., 2017) e (SHEN et al., 2017), algumas medidas são usualmente utilizadas, como preservação do conteúdo, principalmente com a métrica BLEU, e precisão de estilo, a fim de calcular se a frase gerada está no estilo

pretendido. A métrica BLEU mede o quão perto uma sentença está de uma sentença de referência com base em correspondências de n-gramas (PAPINENI, K., 2002). Seu resultado deve ser idealmente próximo de 0,5, pois oscila entre 0 e 1, representando uma frase completamente diferente e uma frase igual, respectivamente, quando comparada a uma frase de referência. Apesar dos avanços obtidos em trabalhos anteriores, a retenção do conteúdo da sentença antes e depois da transferência ainda pode ser melhorada.

Uma abordagem que tem sido cada vez mais bem sucedida na transferência de estilo de texto é o uso de uma Rede Adversária Generativa ou GAN (GOODFELLOW *et al.*, 2014). A estrutura de rede neural é composta por duas redes, uma geradora e outra discriminadora. Essas duas redes são treinadas simultaneamente. A rede geradora tem como objetivo aprender a produzir amostras que correspondam aos dados de treinamento. A rede discriminadora determina se uma amostra é real ou do modelo gerador, dando uma pontuação a cada amostra. Este trabalho adota uma arquitetura de rede neural baseada em Cycle-Gan, uma melhoria ao GAN padrão, que aumenta a preservação de conteúdo durante a transferência de estilo de sentimento, adicionando uma restrição de ciclo no procedimento de treinamento (Huang, Y. *et al.* 2020).

Em geral, o problema que dita as técnicas que compõem o método proposto é a retenção do conteúdo de uma frase gerada após a alteração de seu estilo pretendido. A união dos conceitos que deram forma ao método proposto foi uma combinação de trabalhos anteriores que sugerem melhorar essa retenção de conteúdo, uma vez que estes não exploraram prioritariamente esse conceito. A arquitetura CycleGan, a separação dos espaços latentes e a subsequente aprendizagem supervisionada do estilo espaço latente, contribuem para este objetivo.

A principal contribuição deste trabalho se encontra no método proposto, uma rede neural baseada na arquitetura CycleGan que utiliza um codificador para separar a frase em dois espaços latentes, um para estilo e outro para conteúdo. Durante o processo de treinamento do algoritmo, o espaço latente de estilo foi supervisionado, compreendendo que essa arquitetura e esquema de treinamento ajudariam a melhorar a preservação de conteúdo em um cenário multi-estilo. A adição frente a trabalhos prévios se encontra na utilização combinada da separação dos espaços latentes, juntamente com o esquema de treinamento supervisionado.

1.2 Objetivos

O principal objetivo deste trabalho é o desenvolvimento de um método, baseado na arquitetura CycleGan, capaz de gerar novos textos maximizando a métrica de preservação de conteúdo.

Objetivos específicos:

- Criar um algoritmo capaz de gerar textos com conteúdo semelhante, porém com estilos distintos.
- Comparar o método proposto com métodos disponíveis na literatura.

1.3 Hipótese de Trabalho

As hipóteses desta pesquisa são:

- H1) A separação dos espaços latentes de estilo e conteúdo contribui para a melhoria da métrica BLEU.
- H2) O treinamento supervisionado do espaço latente de estilo contribui para a melhoria da métrica BLEU.
- H3) É possível melhorar a métrica BLEU sem impactar na acurácia na métrica de precisão de estilo.

1.4 Contribuição Científica

A principal contribuição deste trabalho é a criação de uma método capaz de gerar textos com múltiplos estilos, melhorando a preservação do conteúdo do texto antes e depois da transferência. Para tanto, este trabalho reuniu duas abordagens utilizadas anteriormente que melhoraram a métrica de preservação de conteúdo, mas que não tinham sido testadas de forma combinada. A arquitetura proposta separa os espaços latentes de estilo e conteúdo do modelo, além de treinar o espaço latente de estilo de forma supervisionada. Ambas estas adições contribuem

para o desentrelaçamento dos conceitos de conteúdo e estilo, resultando em uma melhor preservação de conteúdo.

1.5 Escopo

O escopo deste projeto de pesquisa se limita a transferência de estilos de textos, com utilização de dados não paralelos e aprendizagem profunda.

1.5 Organização do Documento

O presente documento está organizado em 7 capítulos. No Capítulo 2 é feita uma revisão sobre modelos generativos aplicados a TST. O Capítulo 3 apresenta o estado da arte do tema proposto. Os procedimentos metodológicos utilizados nesta pesquisa são apresentados no Capítulo 4. Um novo método para geração de texto é apresentado no Capítulo 5. O Capítulo 6 apresenta os resultados. Finalmente, o Capítulo 7 conclui o trabalho.

2 Fundamentação Teórica

Este capítulo apresenta os fundamentos teóricos básicos para a compreensão deste trabalho.

2.1 Transferência de estilo de texto

A transferência de estilo de texto (TST) é uma área de pesquisa considerada relativamente nova e foi produto das demandas latentes de duas áreas de pesquisa: transferência neural de estilo (GATYS; ECKER; BETHGE, 2016) e tradução de texto com a utilização de algoritmos de deep learning (BAHDANAU; CHO; BENGIO, 2014).

Como exemplo, uma das formas de transferência de estilo é a transferência de sentimento de texto, cuja execução visa alterar o sentimento subjacente do texto de origem, preservando o conteúdo não sentimental. De forma simples a transferência de sentimento, assim como demonstrado pelas Figuras 1 e 3, visa ensinar um algoritmo a distinguir a dimensão de conteúdo da dimensão de estilo dos textos, alterando somente uma e mantendo a outra inalterada. O conceito de desemaranhamento de dimensões pode ser observado na Figura 2, porém, apesar de nestas figuras este conceito ser abordado para o problema de geração de imagens, a mesma ideia pode ser expandida para textos.

A abordagem comum é construir um modelo de autoencoder que aprende representações latentes desemaranhadas e usa atributos de estilo controláveis para transferir a informação de estilo (HU, Zhiting *et al.*, 2017). Executar a transferência de estilo em uma representação desemaranhada aprendida é um grande desafio e continua sendo um problema de pesquisa em aberto no domínio do texto (FU *et al.*, 2017).

O trabalho (HU, Zhiqiang; LEE; AGGARWAL, 2020) detalhou que várias das técnicas de TST foram adaptadas dos métodos comuns usados em transferência de estilo neural e tradução automática neural. Além disso, algumas das métricas de avaliação usadas no TST também são adaptadas das tarefas de tradução automática neural. A transferência neural de estilo aborda técnicas para adequar

uma imagem, a fim de que adquira as características de estilo artísticas de outra imagem, com a utilização de redes neurais convolucionais (GATYS; ECKER; BETHGE, 2016). Tal técnica é notoriamente aplicada em tarefas de estilização de imagem para que adquiram um estilo específico. Já a tradução automática neural é uma técnica advinda da evolução de algoritmos que anteriormente contavam com modelos estatísticos manualmente definidos para obtenção de uma tradução precisa (BAHDANAU; CHO; BENGIO, 2014). Recentemente estes algoritmos de tradução contam com redes neurais sequenciais para a realização de tal tarefa.

2.1.1 Transferência de estilo neural

O uso de uma rede neural convolucional para a transferência de estilo ocorreu pela primeira vez no trabalho de Gatys e colegas (GATYS; ECKER; BETHGE, 2016), em que os autores abordaram o mesmo conceito de desentrelaçamento de conteúdo e estilo. Este trabalho abriu o novo campo de transferência de estilo neural (TSN), que é o processo de usar redes neurais para formar imagens com conteúdo em diferentes estilos (JING *et al.*, 2020).

Figura 5. Exemplo transferência de estilo neural.
(Extraído de: (GATYS *et al.*, 2016)).

Na Figura 5 pode-se perceber a aplicação prática de TSN, em que a primeira imagem é reescrita com o estilo das imagens menores, criando imagens que retém a mesma paisagem da primeira image, mas com um estilo de pintura diferente.

Matematicamente, a transferência neural de estilo pode ser definida como a mistura do conteúdo de uma imagem (ou texto), com o estilo de referência pretendido. Considere $\vec{x}, \vec{p}\, e\, \vec{a}$ como a imagem que é gerada, a fotografia que é utilizada como fonte e o estilo, respectivamente. A função de perda a ser minimizada é a soma entre o conteúdo e estilo, com determinados pesos $\alpha\, e\, \beta$ que determinam a força da presença de ambos os componentes, como descrito na equação 1.

$$L_{total} = \alpha L_{conteúdo}(p, x) + \beta L_{estílo}(a, x) \quad (1)$$

Existe uma similaridade entre as tarefas de TST e TSN, no que se trata do desentrelaçamento de dimensões latentes a fim de alterar uma de forma independente, sem impactar a outra, e gerar uma nova informação. Apesar desta semelhança, segundo (HU, Zhiqiang; LEE; AGGARWAL, 2020) a tarefa de desentrelaçamento e transferência de estilo em texto é mais complexa por ser mais sutil, o que torna difícil diferenciar e definir estilos em dois ou mais trechos de texto.

2.1.2 Tradução automática neural

A tradução automática neural (TAN), uma abordagem baseada em aprendizagem profunda, é um método bem estudado na área de pesquisa (BAHDANAU; CHO; BENGIO, 2014). Ao contrário das técnicas tradicionais de tradução (PIETRA *et al.*, 1990), o TAN pode realizar treinamento do ciclo completo de tradução automática, sem a necessidade de lidar com alinhamentos de palavras, regras de tradução e algoritmos de decodificação complicados, como os trabalhos tradicionais.

Tanto o TST quanto o TAN são ramos da geração de linguagem natural (GATT; KRAHMER, 2018). Por conta disso, as duas áreas de pesquisa compartilham algumas semelhanças. TAN visa mudar a linguagem de um texto preservando o conteúdo, ao passo que TST visa modificar as propriedades estilísticas de um texto preservando o conteúdo. Ambas as tarefas, apesar de contexto diferentes,

trabalham essencialmente mantendo uma característica do texto e mudando outra, de forma desentrelaçada.

Matematicamente, a tarefa de geração de texto, de modo geral, é definida por uma função de probabilidade condicional que leva em consideração a próxima palavra a ser gerada, y^t, o conjunto de palavras anteriores que compõem a frase $\{y1, \cdots, y_{t-1}\}$ e o contexto da mesma C. Como é possível notar na equação 2, de acordo com (BAHDANAU; CHO; BENGIO, 2014), a descrição do processo de geração de texto pode ser definido como:

$$p(y) = \prod_{t=1}^{T} p(yt, \{y1, \cdots, y_{t-1}\}, c) \quad (2)$$

Por este motivo, a maioria dos modelos de TST são semelhantes, arquiteturalmente, com as técnicas de TAN mais comumente usadas: os modelos de sequência, no formato de codificador-decodificador (CHO *et al.*, 2014). Tais modelos, similarmente como definido pela equação 2, são funções probabilísticas capazes de gerar frases novas.

2.2 Modelo adversário generativo

Dentre os modelos generativos de texto desenvolvidos recentemente, os modelos generativos adversários, conhecidos como GANs (Generative Adversarial Networks), ganharam bastante relevância depois de apresentarem resultados que se assemelham à qualidade humana de geração de dados textuais e imagéticos (alguma ref para confirmar esta afirmação?). Tal relevância se dá também pela proposição de um método de treinamento de uma rede neural sequencial que se assemelha à estratégia de aprendizagem humana, com a presença de um “aluno” e um “professor”.

O propósito central dos algoritmos generativos é a capacidade de aproximar as distribuições de dados sintéticos e reais, criando assim um modelo capaz de compreender, generalizar e criar dados semelhantes aos reais.

Especificamente, as redes generativas adversárias (ou Generative adversarial network - GAN) realizam esta tarefa por meio de um “jogo” entre duas redes neurais. A tarefa da rede principal, comumente chamada de gerador, é aprender a distribuição dos dados reais, a fim de gerar exemplos que confundam o discriminador, a segunda entidade desse sistema, responsável por distinguir dados reais e sintéticos (CHEN, L. *et al.*, 2020).

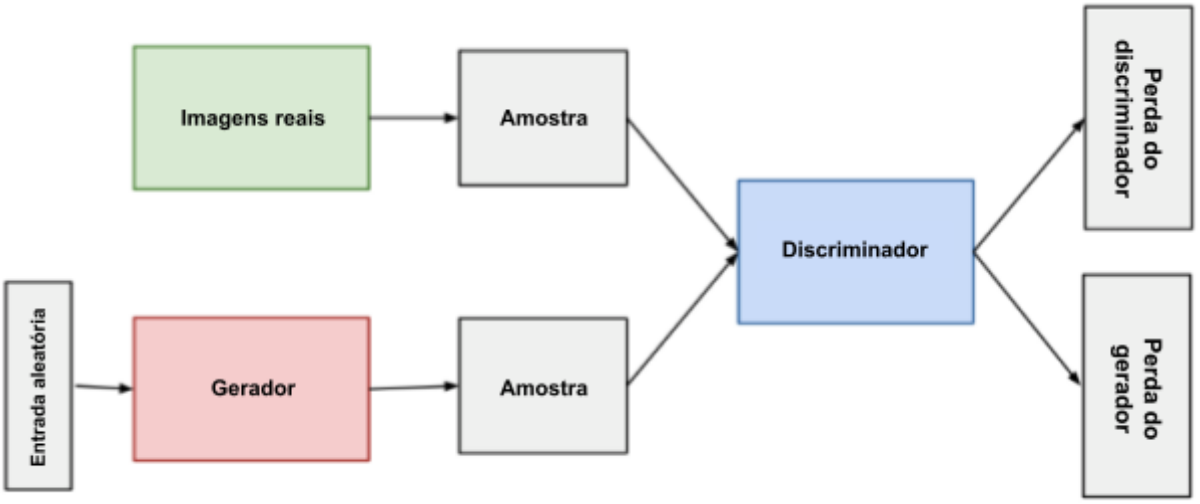

Figura 6. Exemplo de arquitetura de um modelo GAN.
(Imagem adaptada de: https://developers.google.com/machine-learning/gan/images/gan_diagram.svg, em 30/06/2022).

Como pode ser visto na Figura 6, a partir de uma entrada sintética e uma entrada real, o discriminador deve classificar estes dados de forma a conseguir distingui-los. A ideia é que o aumento na assertividade do discriminador é diretamente proporcional à capacidade de distinguir os dados de maneira assertiva, e o aumento na assertividade do gerador ocorre a partir do aumento no erro do discriminador. Intuitivamente, como o discriminador conhece os dados reais e os sintéticos, espera-se que ele consiga distingui-los, e quando isso não acontece significa que o gerador está sendo capaz de gerar dados que se aproximam da distribuição dos dados reais.

Matematicamente, considere $D(x)$ como a estimativa do discriminador da probabilidade de que uma instância real x seja real, e $D(G(z))$ como a estimativa do discriminador da probabilidade de que uma instância falsa z seja falsa. A partir disso,

considerar também que E_x é o valor esperado para todas as instâncias reais e E_z é o valor esperado para todas as instâncias falsas. O objetivo da função geral proposta pelo trabalho de (GOODFELLOW *et al.*, 2014) é que o gerador tente minimizar a seguinte função (3) e o discriminador tente maximizá-la.

$$L_{total} = E_x[log(D(x))] + E_z[log(1 - D(G(z)))] \quad (3)$$

O trabalho de (GOODFELLOW *et al.*, 2014) ainda pontua que, a equação 3 pode não fornecer gradiente suficiente para G aprender bem. No início da aprendizagem, quando G ainda não está otimizada, D pode rejeitar amostras com alta confiança porque são claramente diferentes dos dados de treinamento. Isto ocorre, pois o problema de aprender a gerar dados é mais complexo do que o de somente classificá-los.

Nesse caso, $log(1 - D(G(z)))$ é incapaz de evoluir mais o aprendizado. Em vez de treinar G para minimizar $log(1 - D(G(z)))$ pode-se treinar G para maximizar $log(D(G(z)))$. Esta função objetiva resulta no mesmo ponto fixo da dinâmica de G e D, mas fornece gradientes muito mais fortes no início do aprendizado (GOODFELLOW et al., 2014).

Além desta sugestão para o treinamento, outros problemas já foram constatados nos modelos generativos adversários, como o colapso do modelo e a incapacidade de convergir.

O colapso do modelo ocorre, pois, normalmente, deseja-se que o modelo GAN produza uma ampla variedade de retornos, por exemplo, um rosto diferente para cada entrada aleatória em seu gerador de rostos. Porém, se um gerador produz uma saída especialmente plausível, induzindo o discriminador ao erro, o gerador pode aprender a produzir apenas essa saída (METZ *et al.*, 2016).

Se o gerador começar a produzir a mesma saída repetidamente, a melhor estratégia do discriminador é aprender a sempre rejeitar essa saída. Mas se a próxima geração de discriminador ficar presa em um mínimo local e não encontrar a melhor estratégia, então é muito fácil para a próxima iteração do gerador encontrar a saída mais plausível para o discriminador.

Em relação à incapacidade de convergir, (ARJOVSKY; BOTTOU, 2017) atestaram que se o discriminador for muito bom, o treinamento do gerador pode

falhar devido ao desaparecimento dos gradientes. Um discriminador ideal não fornece informações suficientes para o gerador fazer progresso.

Portanto, o treinamento de um modelo GAN é uma tarefa complexa que requer inúmeros testes para se encontrar uma metodologia de treinamento que resolva o problema em questão.

2.3 CycleGAN

Uma metodologia de treinamento introduzida por (ZHU *et al.*, 2017) é o CycleGAN, que trata o problema de transferência de estilo entre dados não pareados com um mecanismo de ciclo. A ideia é que uma vez transferidos os dados para o estilo latente desejado, o modelo deve ser capaz de retorná-los para o estilo original sem grandes mudanças. Na Figura 7 pode-se ver este ciclo explicitamente pela função de perda L2, em que o processo de treinamento compara a imagem original e a imagem reconstruída para garantir a consistência da transferência.

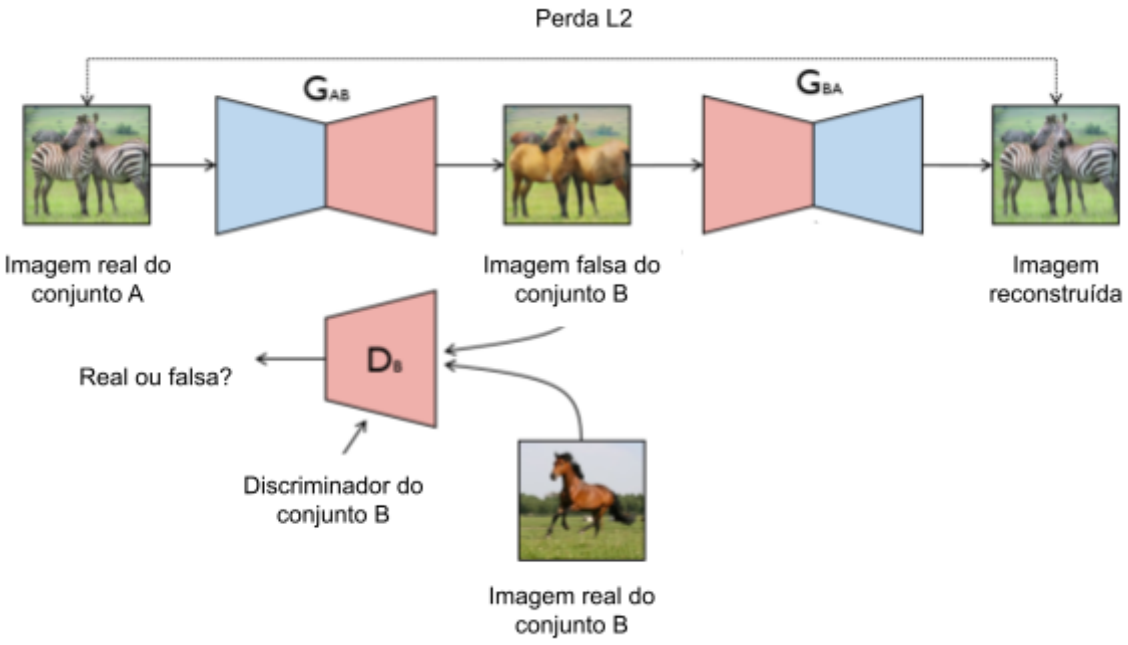

Figura 7. Exemplo de arquitetura de um modelo CycleGAN.
(Imagem adaptada de: https://blog.jaysinha.me/train-your-first-cyclegan-for-image-to-image-translation/, em 30/06/2022).

Apesar de o conceito de CycleGAN ter sido desenvolvido no contexto de imagens, alguns trabalhos foram capazes de utilizar o mesmo conceito para TST, como (IDA *et al.*, 2020). Em seu trabalho os autores foram capazes de adaptar um modelo de sequência para sequência, utilizando a consistência de ciclo, obtendo resultados superiores à modelos anteriores no dataset *Yelp Review*[1]. Apesar disso, os autores sugerem que a utilização de modelos GAN e consistência de ciclo devem ser melhor explorados, dada a complexidade da tarefa.

Diversos são os trabalhos desenvolvidos que visam evoluir a pesquisa em modelos generativos, principalmente no que diz respeito a novas arquiteturas de redes neurais capazes de modelar este problema de forma mais assertiva, mas também quanto ao contexto de aplicação que o algoritmo se propõe a atuar. Desta forma, o terceiro capítulo de estado da arte visa explorar principalmente duas perguntas em relação a este cenário apresentado acima. Uma no que tange às arquiteturas e modelos mais aplicados para o problema de geração de texto e outra para os problemas e aplicações práticas que estes se propõem a resolver.

[1] https://www.yelp.com/dataset

3 Estado da Arte

Neste capítulo são apresentados os trabalhos relevantes relacionados a esta pesquisa: transferência de estilo de textos não paralelos. Para reunir os trabalhos mais relevantes para a pesquisa, foram utilizadas as bases IEEE[2], Scopus[3], ACM[4], Springer[5] e ScienceDirect[6].

A *string* de busca utilizada para criar tal seleção de trabalhos contou principalmente com os seguintes termos “text”, “non-parallel” e “style transfer”, visto que estas são as palavras que mais distinguem e filtram os trabalhos a serem encontrados. Além disso, percebeu-se alguma interferência de trabalhos que realizam transferência de estilo em outros tipos de dados como imagens e áudio, portanto, retirou-se estes termos da pesquisa e filtrou-se a busca somente a partir do título, abstract e palavras-chave.

A busca nas bases citadas anteriormente, que utilizou a *string* de busca, retornou 17 trabalhos no total, que, após a exclusão dos repetidos, ficou com 12. A string de busca utilizada para encontrar os trabalhos que serviram de base para determinar o estado da arte atual variou dependendo da sintaxe da base de trabalhos utilizada, mas no geral seguiu o seguinte formato:

("Abstract":text OR "Abstract":sentence) AND ("Abstract":non-parallel) AND ("Abstract":style transfer) NOT ("Abstract":image OR "Abstract": vídeo OR "Abstract": audio OR "Abstract": sound).

A função prática desta coleta foi verificar e resumir os aspectos considerados mais importantes para o avanço da transferência de estilo de texto não-paralelos, e, na prática, respondeu a duas perguntas de pesquisa. A primeira pergunta da pesquisa foi em relação aos estilos utilizados pelos trabalhos, especificamente em relação a quais são os estilos mais utilizados e qual o número de estilos utilizados para transferência. Esta pergunta foi inspirada pelo trabalho de (HU, Zhiqiang; LEE; AGGARWAL, 2020), em que os autores salientam esta lacuna de pesquisa. A segunda pergunta visou detalhar e explorar quais arquiteturas de rede neural foram mais utilizadas por trabalhos anteriores.

---

[2] https://ieeexplore.ieee.org/
[3] https://www.scopus.com/
[4] https://dl.acm.org/
[5] https://link.springer.com/
[6] https://www.sciencedirect.com/

Para responder a primeira pergunta de pesquisa, a Tabela 1 organiza cada um dos trabalhos coletados, apresentando duas características sobre cada um: o número de estilos utilizados pelo trabalho para transferência e quais foram estes estilos.

Tabela 1. Estilos utilizados para transferência por trabalhos anteriores.

Artigo	**Número de Estilos**	**Nomes dos Estilos**
(CHEN, X. *et al.*, 2021)	3	chinês simplificado, chinês tradicional e cantonês
(SHEN *et al.*, 2017)	2	Negativo, positivo
(CHEN, L. *et al.*, 2020)	2	Negativo, positivo
(SANTOS *et al.*, 2018)	2	Ofensivo, não ofensivo
(HU, M.; HE, 2021)	2	Negativo, positivo
(LAI *et al.*, 2019)	4	Negativo, positivo, pretérito, presente
(LI *et al.*, 2019)	4	Negativo, positivo, formal , informal
(IDA AYU PUTU; LEE, 2020)	2	Negativo, positivo
(KAWASHIMA; TAKAGI, 2019)	2	Complex, simple
(HAN; WU; NIU, 2018)	2	Shakespeare, modern English
(CHEN, J. *et al.*, 2019)	2	Negativo, positivo
(PENG *et al.*, 2019)	2	Notícia, poema

É relevante notar que a maioria dos trabalhos utiliza a validação com o corpus de inversão de polaridade de sentimento do Yelp ou Amazon, e, sendo assim, estão trabalhando com uma transferência de estilo com somente duas possibilidades. Este padrão percebido norteou uma das principais contribuições deste trabalho, a criação de uma algoritmo que aprenda a transferir textos entre múltiplos estilos.

Para responder a segunda pergunta de pesquisa, a Tabela 2 traz as arquiteturas utilizadas por cada trabalho para realizar a transferência de estilo.

Tabela 2. Arquiteturas de redes neurais utilizadas.

Artigo	**Arquitetura Utilizada**
(CHEN, X. *et al.*, 2021)	Non-Autoregressive Transformer (NAT)
(SHEN *et al.*, 2017)	Cross-aligned auto-encoder (CAE)
(CHEN, L. *et al.*, 2020)	Feature mover GAN (FM-GAN)
(SANTOS *et al.*, 2018)	Neural Text Style Transfer
(HU, M.; HE, 2021)	Deterministic autoencoder (DAE)
(LAI *et al.*, 2019)	Generative adversarial network (GAN)
(LI *et al.*, 2019)	Domain Adaptive Style Transfer (DAST)
(IDA AYU PUTU; LEE, 2020)	Sequence to sequence CycleGAN
(KAWASHIMA; TAKAGI, 2019)	Generative adversarial network (GAN)
(HAN; WU; NIU, 2018)	Text Style Transfer Seq2Seq
(CHEN, J. *et al.*, 2019)	Conditional GAN
(PENG *et al.*, 2019)	Sequence to sequence CycleGAN

As arquiteturas propostas nos trabalhos em geral utilizam em sua totalidade algoritmos auto codificadores, de sequência para sequência, dada a natureza do problema a ser resolvido, uma frase de entrada no modelo e uma frase de saída. Os dois modelos generativos mais utilizados como base teórica nos trabalhos foram os Auto-codificadores Variacionais (VAEs) e as Redes Generativas Adversárias (GANs). Os trabalhos analisados propuseram novas nomenclaturas para suas arquiteturas, por conta de alguma melhoria ou personalização sugerida pelos autores para aperfeiçoamento da performance em diversos aspectos.

Um modelo que foi notoriamente utilizado pela maioria dos autores foi o CycleGAN, que originalmente foi utilizado para problemas de transferência de uma imagem para outra imagem *“image-to-image translation”*. O método desenvolvido pelo trabalho (ZHU *et al.*, 2017) avançou a pesquisa para modelos generativos sem a necessidade de bases de dados pareadas, dada a dificuldade de obtenção destas na maioria dos casos. Este modelo tenta aprender um mapeamento sem exigir dados emparelhados, usando redes adversárias mais consistentes por meio de um mecanismo de ciclo.

A contribuição trazida pela Tabela 2 foi o direcionamento do tipo de arquitetura que mais tem sido utilizado por trabalhos que desenvolveram algoritmos de transferência de estilo com bases não pareadas.

Como apresentado anteriormente, para determinar a acurácia dos modelos, trabalhos anteriores utilizaram métricas que visam medir a acurácia na transferência para o estilo pretendido, e a preservação do conteúdo da frase original. Esta última é utilziada para garantir que a frase não seja muito diferente nem igual à frase original. A pontuação BLEU é a principal métrica usada para determinar quanto de uma frase transferida reteve seu conteúdo original, portanto, usado por trabalhos de transferência de estilo para determinar a precisão parcial do modelo. A interpretação do escore BLEU, que oscila entre 0 e 1, sugere que, para a tarefa de transferência de estilo, o valor ideal estaria em algum lugar no meio de seu intervalo, visto que a frase não teria sido nem muito alterada, nem teria permanecido igual (SCHMIDT; BRAUN, 2022).

Por exemplo, tendo a frase de base como sendo: “Eu amo o Brasil e o seu povo”, se compararmos com a mesma frase, o resultado seria 1.0. Se alterássemos a frase de comparação para: “Eu amo o Paraguai e o seu povo”, o resultado do cálculo seria 0.5. Por fim, se alterássemos a frase de comparação para: “Eu adoro o Paraguai e o seu povo”, o resultado do cálculo seria 0.38, mostrando assim que quanto mais altera-se a frase, menos é o resultado do cálculo.

Tratando agora especificamente de trabalhos que realizaram transferências entre múltiplos estilos, comparando três trabalhos utilizados por (LAI et al., 2019) como forma de comparação, os modelos desenvolvidos pelos trabalhos (LOGESWARAN, LEE, BENGIO, 2018) e (LAI et al., 2019) resultaram em uma média de 27,66% para a métrica BLEU, sendo um resultado inferior ao ideal (SCHMIDT; BRAUN, 2022). Isto se dá pois segundo (SCHMIDT; BRAUN, 2022) e (JAIN *et al.*, 2019) com valores próximos a 1 falha em aprender diferenças de estilo suficientes e aqueles próximos a 0 distantes do conteúdo da frase original.

Segundo (LAI et al., 2019), responsável pela criação de um método que melhora a preservação de conteúdo em relação a trabalhos anteriores, dois aspectos são centrais nos resultados obtidos pelos mesmos. O primeiro aspecto é o treinamento dividido em duas etapas, a primeira retirando a perda adversária e somente incentivando o modelo a aprender a reconstruir o texto. Na sequência é

adicionada também a perda adversária, garantindo assim, primeiramente, que o mesmo aprenda a reconstruir a sentença para depois modificá-la.

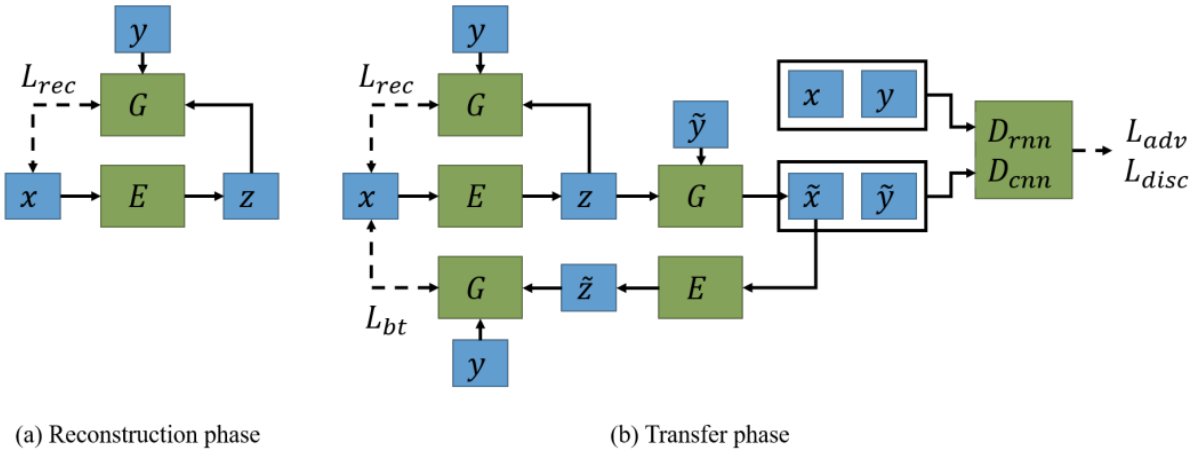

Figura 8. Exemplo de arquitetura proposta por LAI.
(Adaptado de:: https://aclanthology.org/D19-1366.pdf).

Na Figura 8, nota-se a presença de quatro funções de perda distintas, a função de reconstrução, de *back translation,* adversária e do discriminador. É interessante ressaltar aqui que a função de *back translation* é análoga a função de ciclo nas arquiteturas CycleGan. Durante a primeira etapa de treinamento somente a L_{rec} foi otimizada, visando garantir primeiramente que o modelo aprenda a reconstruir o texto, contribuindo também para uma melhoria na retenção do conteúdo, segundo os autores. Na segunda etapa, além da função de reconstrução, todas as outras foram levadas em consideração. Este processo serviu de inspiração para o nosso trabalho e foi aplicado de forma análoga durante o processo de treinamento.

Outro trabalho (LIN *et al.*, 2020) sugeriu uma abordagem para melhorar a preservação de conteúdo, trazendo a ideia de separação de espaços latentes. A estrutura proposta criou dois codificadores, cada um mapeando a frase original para espaços latentes de estilo e conteúdo, respectivamente. O espaço de estilo durante o treinamento é incentivado de forma supervisionada a aproximar o estilo original da frase. Posteriormente ambos os espaços latentes foram então combinados e passados pelo decodificador para reconstrução da sentença. Os autores relataram que tal incentivo supervisionado, juntamente com a separação dos espaços em meio

a arquitetura, contribuíram para a retenção do conteúdo do texto após uma transferência de estilo. Esta referência sugeriu um dos principais conceitos aplicados por este trabalho, o de separar os espaços latentes de estilo e conteúdo, treinando de forma supervisionada o de estilo.

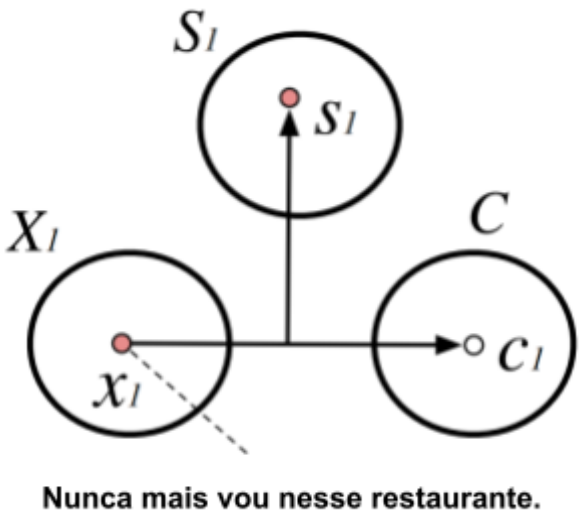

Figura 9. Exemplo de arquitetura proposta por Lin.
(Adaptado de: https://arxiv.org/pdf/2002.06525.pdf).

Na Figura 9 podemos ver um esquema do método proposto por (LIN *et al.*, 2020), em que um texto original (X) é mapeado em dois espaços distintos de conteúdo e estilo.

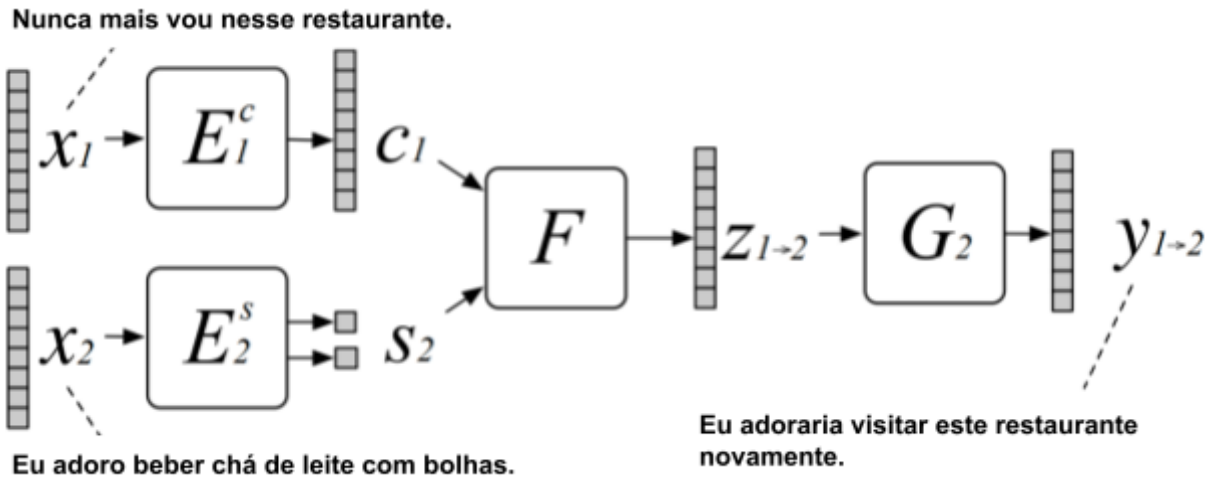

Figura 10. Exemplo de arquitetura proposta por Lin.
(Adaptado de: https://arxiv.org/pdf/2002.06525.pdf).

Na Figura 10 podemos compreender melhor o mecanismo de funcionamento deste algoritmo. Os autores primeiramente passam uma frase (X1) pelo codificador, e guardam seu espaço latente de conteúdo. Após, os autores passam uma frase (X2) positiva pelo codificador, mas somente guardam seu espaço latente de estilo. Por fim, ambos os espaços são combinados e passados ao gerador ou decodificador, que gera uma frase similar a frase X1, porém com o estilo da frase X2 (positivo).

Por fim, de forma geral, a análise desenvolvida no capítulo de estado da arte contribuiu com a resposta de duas perguntas de pesquisa, uma em relação aos estilos contemplados pelos modelos de transferência de estilo, e outra em relação às arquiteturas mais utilizadas pelos autores. A contribuição trazida neste capítulo foi a de desenvolver um algoritmo capaz de trabalhar com múltiplos estilos. Este capítulo sugere que a arquitetura que tem sido mais utilizada pelos autores é a CycleGAN, que, apesar de ter sido desenvolvida inicialmente para problemas que envolvessem imagens, com algumas adaptações pode contemplar problemas envolvendo texto e, portanto, a transferência entre estilos de textos.

Apesar de existirem trabalhos prévios que tratam do tema de transferência de estilo, nenhum desenvolveu um método como foco na melhora da preservação de conteúdo, a maioria focou na melhora da precisão do estilo. A preservação é um tópico tão importante quanto a precisão do estilo, visto que de nada adianta um método que realize a transferência, mas que para tanto precise alterar muito a sentença original. Para tanto, os principais direcionamentos no sentido de arquitetura foram de separar os espaços latentes de estilo e conteúdo, além de treinar o espaço de estilo de forma supervisionada. Durante o treinamento, outra orientação é a de separar o processo em duas etapas, por sugerir uma melhoria na preservação do conteúdo após uma transferência.

4 Procedimentos Metodológicos

Este capítulo apresenta os procedimentos metodológicos utilizados nesta pesquisa. A estrutura desta pesquisa é dividida em quatro partes: Planejamento inicial, Fase exploratória, Desenvolvimento e Avaliação.

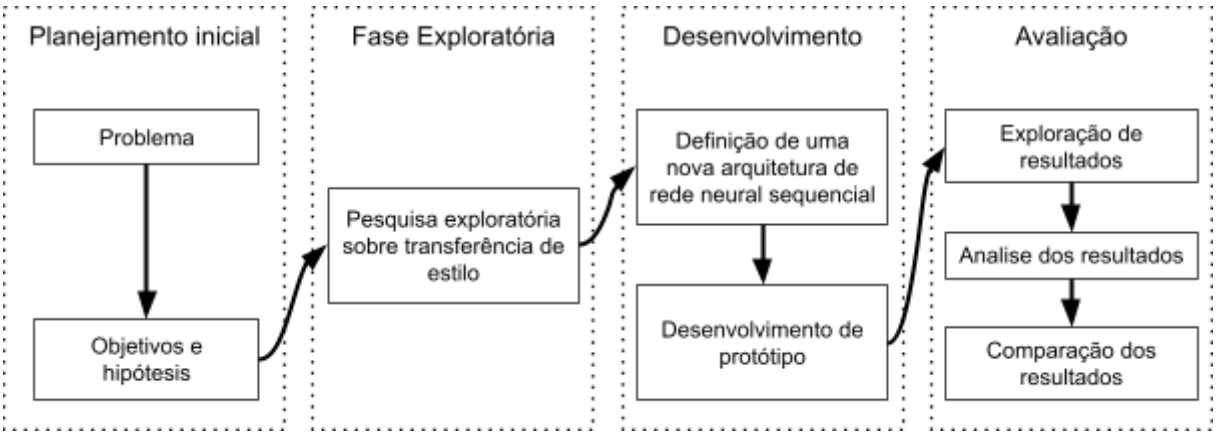

Figura 11. Estrutura da pesquisa.

As seções na sequência apresentam as etapas apresentadas na Figura 11 com maior detalhe.

4.1 Planejamento Inicial

Analisando as tendências de algoritmos e aplicações com lacunas latentes, constatou-se que a aplicação de técnicas de transferência de estilo em texto ainda possuem um grande potencial de desenvolvimento e evolução, principalmente no que diz respeito a criação de um método que seja capaz de gerar resultados que melhorem a preservação de conteúdo, por meio do cálculo da métrica BLEU.

De forma a realizar e evoluir a pesquisa neste sentido, faz-se necessária a exploração de técnicas previamente desenvolvidas que pudessem contribuir para o desenvolvimento deste trabalho.

O principal objetivo deste trabalho é o desenvolvimento de um método, baseado na arquitetura CycleGAN, capaz de gerar textos melhorando a métrica de preservação de conteúdo em relação a trabalhos prévios.

De forma geral, a arquitetura generativa de rede neural utilizada foi o CycleGan, introduzida por (ZHU *et al.*, 2017), e, por se tratar de uma aplicação de transferência de estilo, em que se pretende alterar o estilo sem modificar de forma significativa o conteúdo, o primeiro objetivo específico definido para este trabalho foi desenvolver um algoritmo responsável por gerar textos com conteúdo semelhante, porém com estilos distintos.

O segundo objetivo específico foi o de comparar o método proposto com métodos disponíveis na literatura, a fim de investigar sua performance quando comparado aos modelos anteriores, principalmente no que diz respeito à preservação de conteúdo antes e após uma transferência de estilo.

4.2 Fase exploratória

Durante a fase exploratória realizou-se uma pesquisa sobre o uso de modelos generativos de texto para transferência de estilo. Essa etapa ocorreu para fornecer embasamento para que a sequência do trabalho possuísse consistência e relevância frente ao que já foi desenvolvido.

A pesquisa desses trabalhos foi realizada por uma revisão da literatura, como definido pelo Capítulo 3 de Estado da Arte, e resultou na compreensão de quais arquiteturas de rede neural são as mais utilizadas para este problema de transferência de estilo. Além disso, foram explorados padrões de arquitetura e treinamento utilizados por trabalhos prévios, que contribuíram para os objetivos deste trabalho.

4.3 Desenvolvimento

A fim de desenvolver uma nova arquitetura de rede neural para transferência entre múltiplos estilos, diversas arquiteturas foram analisadas para determinar quais trabalhos previamente publicados poderiam ser adaptados e utilizados. Com base em arquiteturas previamente desenvolvidas para transferência de estilo, notou-se que a maioria possui a capacidade de transferência entre dois estilos, sendo incompatível com a proposta deste trabalho.

As bases de dados utilizadas para compreensão de diferentes estilos e visando comparação com trabalhos prévios, foram as bases AmaProd e YelpTense Dataset, tendo como referência os trabalhos de (RUINING; MCAULEY; 2017) e (LAI et al., 2019).

Além disso, durante o treinamento do modelo, como descrito por (LAI et al., 2019), dois otimizadores distintos devem ser utilizados para que as diferentes partes da rede generativa evoluam de forma separada. Um otimizador é responsável pela evolução da parte generativa de maneira geral do algoritmo e o outro otimizador é responsável pela evolução da parte adversária. Isto se dá para que os pesos corretos da rede neural sejam otimizados nas funções de perda corretas. As definições matemáticas formais de tais cálculos podem ser consultadas no Capítulo 5.

Durante o treinamento as bases de dados foram divididas entre treino e validação, a fim de medir a estabilidade do treino durante o processo em dados não conhecidos pelo algoritmo. Utilizou-se 20% dos dados para o split de validação.

4.4 Avaliação

Como descrito no primeiro capítulo, a fim de avaliar os objetivos específicos deste trabalho, algumas métricas serão utilizadas para medir os resultados obtidos, assim como algumas comparações para se constatar as diferenças de performance frente a resultados anteriormente expostos por outros estudos.

Para o primeiro objetivo específico de gerar conteúdos semelhantes, com estilos diferentes, assim como sugerido por trabalhos anteriores (HU, M.; HE, 2021), a métrica BLEU será utilizada.

Para o cálculo da métrica BLEU, assim como descrito por (PAPINENI *et al.*, 2002), é necessário a construção de um modelo de n-gramas entre as frases com estilo transferido e as frases originais, então utiliza-se os mesmos n-gramas para avaliar sua similaridade, assim como descrito na equação 5.

$$BLEU = BP \cdot exp(\sum_{n=1}^{N} w_n \log p_n) \quad (5)$$

A seguir tr(x) será utilizado como sinônimo da frase resultante de uma transferência de estilo. Na equação 5, BP é a penalidade de brevidade para as somas de palavras entre *x* e tr(x), e p_n é a razão entre as mesmas contagens de n-gramas em x e as contagens de n-gramas em tr(x) (N é geralmente definido como 3). W_n representa os pesos que somam 1. A adaptação do algoritmo aplicado por este trabalho[7] utilizada a versão percentual do cálculo.

Durante o treinamento, a fim de avaliar de forma mais precisa o algoritmo, a base de dados foi dividida aleatoriamente 75% para treinamento, e 25% para teste, levando em consideração uma igual distribuição de rótulos. Tal processo se deu a fim de medir se o algoritmo estaria de fato aprendendo a generalizar o aprendizado, ou estaria simplesmente decorando os dados.

A fim de consolidar comparações com trabalhos prévios, as bases de dados utilizadas por este trabalho são da Yelp e da Amazon, e tratam dos contextos diferentes. A primeira trata da inversão de sentimentos e tempo verbal da frase, ao passo que a segunda trata da inversão do sentimento e da alteração do produto em questão na sentença. Maiores detalhes sobre as bases foram apresentados no capítulo de resultados do trabalho.

[7] https://github.com/zhijing-jin/bleu

5 Método proposto

Este capítulo apresenta um método para transferência de estilo utilizando CycleGan. A arquitetura proposta pode ser definida formalmente com uma CycleGan, porém com o acréscimo de separar os espaços latentes de estilo e conteúdo, treinar de forma supervisionada o espaço de estilo e dividir o treinamento do algoritmo em duas etapas.

5.1 Pressupostos

De acordo com o objetivo do trabalho, a proposição do método parte do pressuposto que este deve ser capaz de transferir para mais de dois estilos. Dada a complexidade, e a característica esparsa dos dados utilizados, visto que são textos, foram utilizadas técnicas de Aprendizagem Profunda para o desenvolvimento do método.

Para que o algoritmo opere com o conceito de geração condicional, e leve em conta estes *N* estilos de redação no momento de gerar a frase, sua entrada prevê o fornecimento de dois vetores, um obtido com o método *one-hot-embedding* com as palavras que compõem a frase, e um vetor também *one-hot* de *N* posições representando os estilos. Ao longo do treinamento do algoritmo, este é incentivado a conseguir compor frases com diferentes combinações de estilo, por meio de uma representação *one-hot* do número de estilos possíveis.

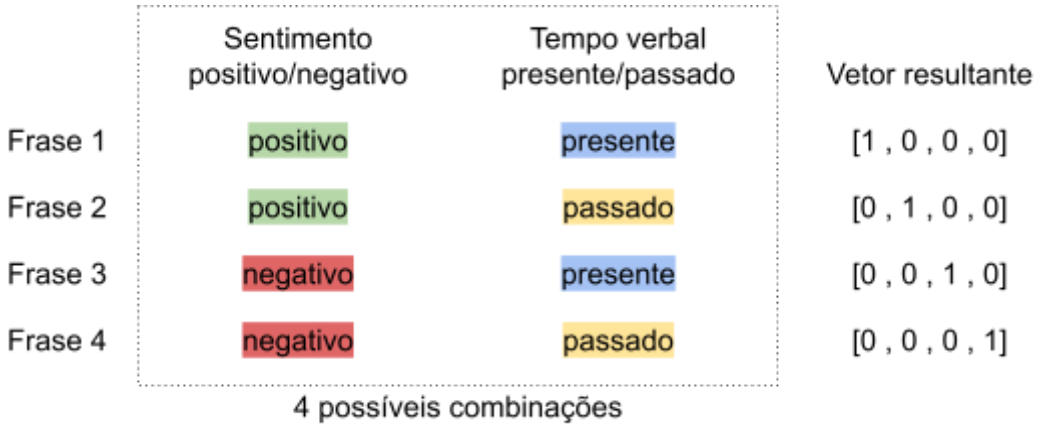

Figura 12. Criação do vetor de estilo.

Assim como visto na Figura 12, utilizando o exemplo da base de dados YelpTense, para criar os vetores *one-hot* de estilo, o vetor resultante possui o tamanho do número de composições possíveis. Tal método de codificação é utilizado por (LAI et al., 2019) e implica em que cada vetor representa mais de uma dimensão de alteração. Por exemplo, um vetor que possua a primeira posição igual a 1 e as demais zero, representa uma frase com estilo resultante positivo e no presente, já um vetor com a quarta posição 1 e as demais zero, resultará em uma frase com estilo negativo e no passado.

5.2 Arquitetura

Conforme discutido no capítulo de estado da arte, como o método proposto se trata de uma rede generativa adversária, duas arquiteturas de rede neural devem ser desenvolvidas, uma geradora e uma discriminadora, responsáveis pela geração das frases propriamente ditas e discriminação das frases geradas e reais, respectivamente.

O método proposto é baseado no CycleGan (Huang, Y. *et al.* 2020), utilizando uma arquitetura no formato codificador-decodificador, separando os espaços latentes criados pelo codificador em um espaço responsável pelo conteúdo z_c e outro para o estilo z_s. Geralmente, os principais conceitos que orientam o método começam com o texto bruto e, antes de serem inseridos no modelo, as frases são tokenizadas e convertidas para o formato *hot*, então passadas pelo Codificador (E). Em seguida, os dados são concatenados e passados pelo modelo Decodificador (G) para retorná-los ao formato original da frase. Por fim, para adicionar a parte adversária ao modelo, durante o treinamento, a arquitetura também é otimizada para minimizar a distância do valor atribuído pelo discriminador (D) às frases reais e geradas pelo decodificador. Uma representação visual do esquema geral do fluxo de dados da arquitetura pode ser acompanhada na Figura 13.

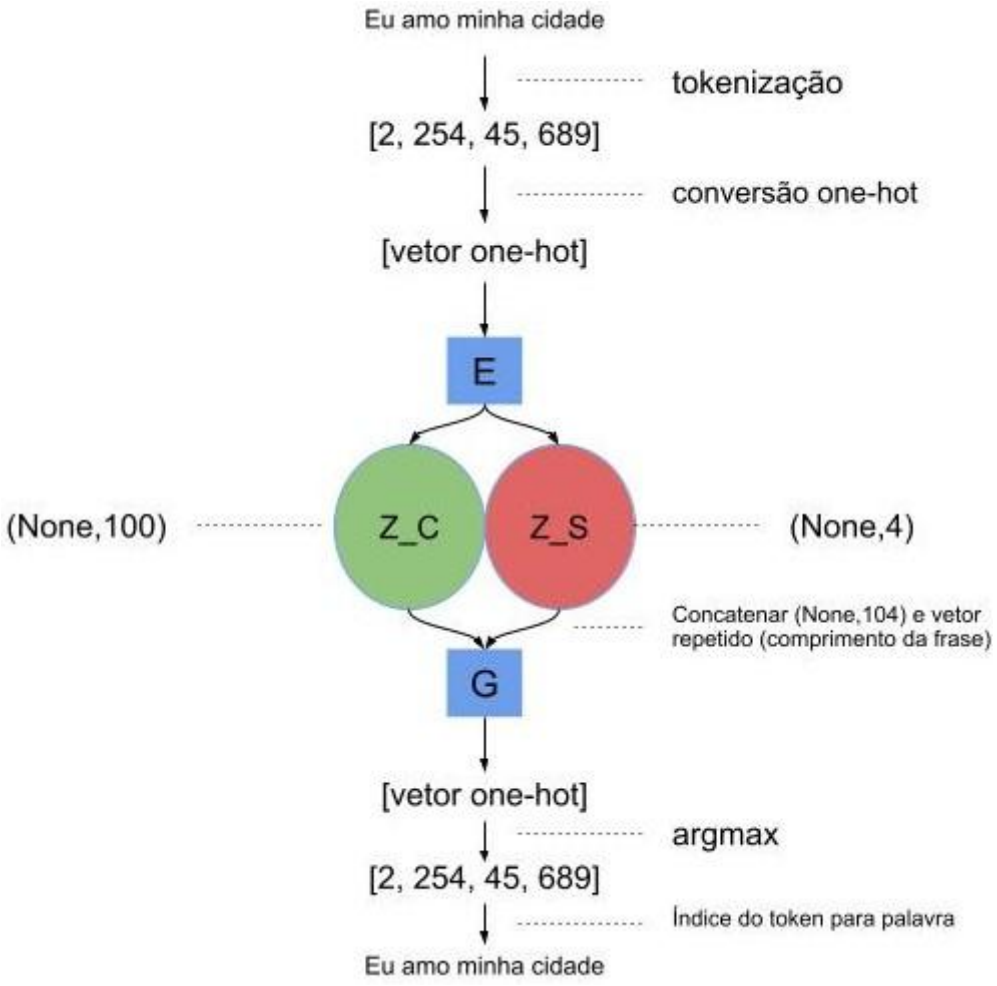

Figura 13. Fluxo de dados.

Na Figura 13 pode-se compreender o fluxo de dados do modelo de uma maneira geral. As frases antes de serem fornecidas como entrada são tokenizadas, e depois transferidas para uma representação *one-hot*. Neste primeiro processo de tokenização é criado um número para cada palavra, pelo qual as mesmas são substituídas. Na sequência, o processo de criação do *one-hot* utiliza estes valores para criação de um vetor em que somente o índice que representa o valor da palavra possui o valor 1, contendo 0 em todas as outras posições. Estes vetores *one-hot* possuem *N* posições, sendo *N* o tamanho do vocabulário. Após as passagens pelo algoritmo, ocorre o processo contrário ao de criação dos vetores *one-hots*, chamado de argmax, em que retorna-se o índice do vetor que possui valor igual a 1. Por fim, tendo os valores das palavras, substitui-se novamente os números pelas palavras que representam, reconstruindo assim a frase original.

Neste ponto, é importante destacar que, apesar do exemplo apresentado na Figura 13 estar em inglês, este algoritmo independe de língua.

Na sequência, na Figura 14, pode-se compreender a composição geral do método, levando em consideração os fluxos de dados internos, relações entre os blocos da rede neural e funções de perda. De acordo com o objetivo do trabalho, tendo como foco a preservação de conteúdo do texto após uma transferência, a Figura 14 deixa claro as duas adições feitas com tal objetivo, a separação dos espaços latentes de estilo e conteúdo (z_s e z_c), além do treinamento supervisionado de z_s. Tal figura servirá de base para as demais sessões, como forma de descrever, detalhadamente, todos os processos envolvidos.

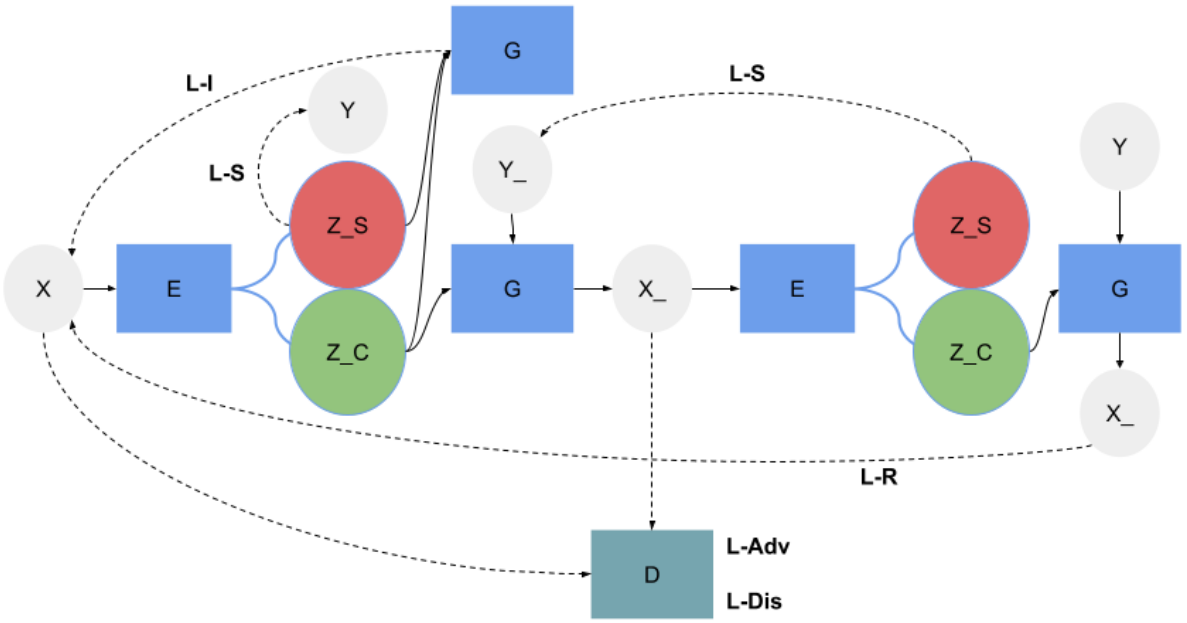

Figura 14. Explicação visual do modelo.

É importante citar que o processo descrito na Figura 14 descreve o treinamento do modelo. Após o treinamento somente a capacidade do Codificador (E) e Decodificador (G) são utilizadas para geração de frases com estilo transferido.

5.2.1 Codificador

Seguindo os trabalhos anteriores, foi utilizado um codificador-decodificador seq-to-seq (E, G). Para cada sentença de entrada real x e rótulo y, o codificador E codifica x em dois espaços latentes com o código $[z_c, z_s] = E(x)$. A separação dos

espaços latentes é dada para posterior aprendizagem supervisionada do espaço latente do estilo, para então ser novamente concatenado com o conteúdo. Depois de concatenados, os dois espaços latentes tornam-se entradas do G, sendo $\tilde{x} = G([z_c, z_s])$.

Em se tratando da sequência de layers utilizados no codificador, seguindo o trabalho de (LAI *et al.*, 2019), foram utilizadas duas camadas GRU com 300 neurônios cada, seguido de uma divisão com mais duas camadas GRU, resultando nos dois espaços latentes de tamanho 100 e 4, para o conteúdo e estilo respectivamente. É importante ressaltar aqui que o 4 está sendo utilizado como exemplo, mas que dependendo do número de estilo da base de dados utilizada este valor pode sofrer alterações. Uma representação da sequência de layers utilizados no codificador pode ser visualizado na Figura 15.

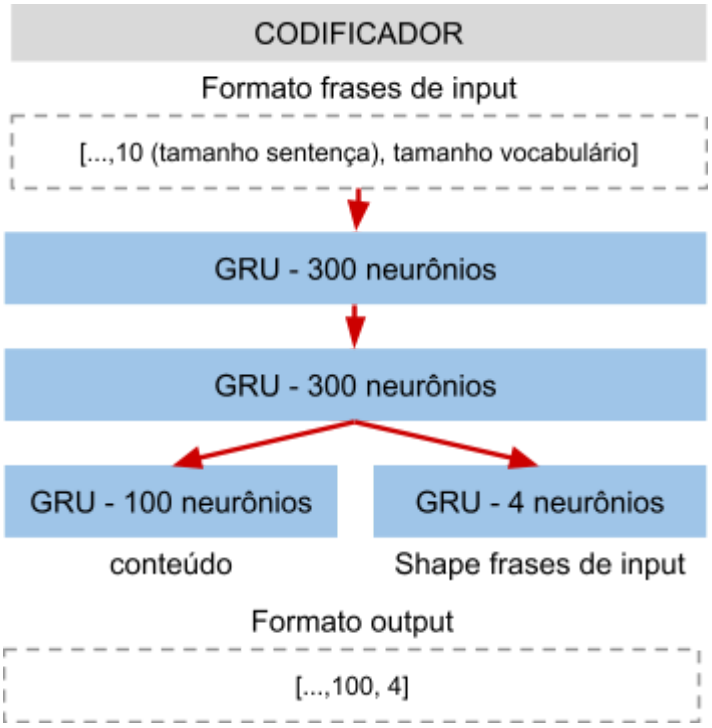

Figura 15. Configuração das camadas do codificador.

Já a sequência de layers utilizados no decodificador, seguindo o trabalho de (LAI *et al.*, 2019), inicialmente o output do codificador é concatenado e repetido pelo tamanho da frase. Na sequência foram utilizadas duas camadas GRU com 300 neurônios cada, por fim, uma camada GRU com o tamanho do vocabulário a fim de aproximar um vetor one-hot, com as probabilidades da palavras que irão compor a

sentença em dada posição. Uma representação da sequência de layers utilizados no decodificador pode ser visualizado na Figura 16.

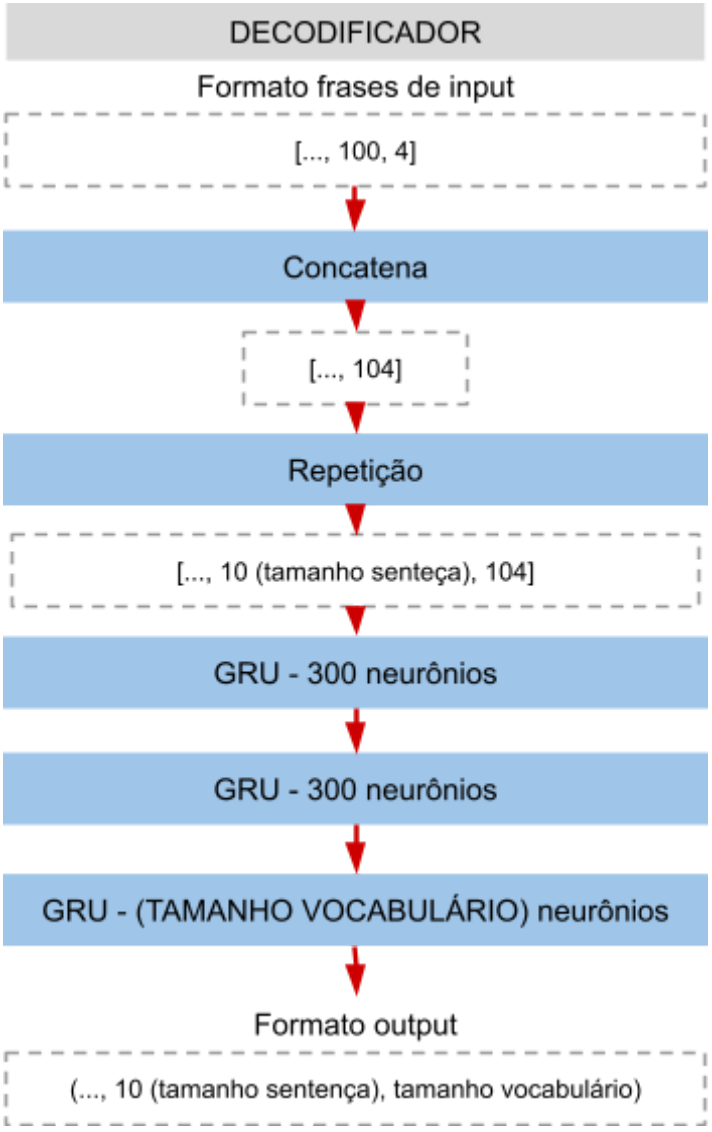

Figura 16. Configuração das camadas do decodificador.

5.2.2 Discriminador

O discriminador toma a representação *hot* como entrada junto com os rótulos de atributos reais y. Formalmente, o discriminador D é definido como $g = D(\tilde{x}, y)$, onde g é a nota de saída do discriminador mapeado por uma função sigmóide.

Na sequência de layers utilizados no codificador, seguindo o trabalho de (LAI *et al.*, 2019), foram utilizadas duas camadas GRU com 300 neurônios cada,

resultando nos dois espaços latentes de tamanho 100 e 4, para o conteúdo e estilo respectivamente.

Em se tratando da sequência de layers utilizados no discriminador, seguindo também o trabalho de (LAI *et al.*, 2019), foram utilizadas duas camadas GRU com 300 neurônios cada, e uma camada densa com um neurônio, resultando em somente um valor de output. Isto se dá visto que o discriminador é projetado para dar notas às sentenças, atribuindo valores maiores a sentenças reais e menor as falsas. Uma representação da sequência de layers utilizados no discriminador pode ser visualizado na Figura 17.

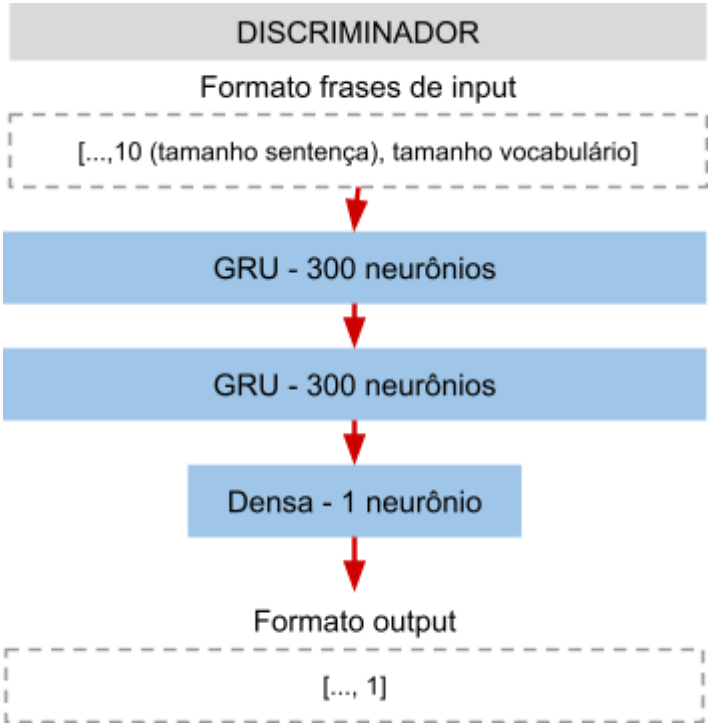

Figura 17. Configuração das camadas do discriminador.

5.2.3 Funções de perda

No início do treinamento, como visto na Figura 14, a versão *hot* codificada da frase original é fornecida como a entrada de E, resultando em dois espaços latentes, um para o estilo z_{s1} e um para o conteúdo z_{c1}. Nesta fase, a primeira de um total de duas perdas de estilo L_s é calculada, comparando o espaço de estilo latente com o vetor de estilo original com uma perda de entropia cruzada padrão, como visto na equação 6.

$$L_{s1} = \quad -\sum y * log\, \bar{z}_{s1} \text{ (6)}$$

Em seguida, ambos os espaços latentes (z_{c1}, z_{s1}) são concatenados para a primeira transformação, retornando a frase para a versão *hot* codificar passando-os pelo G, como visto na equação 7. Essa etapa contribui para o aprendizado da função de identidade L_{id}, que garante que o modelo aprenda a simplesmente gerar a frase fornecida como entrada.

$$L_{id} = \quad -\sum x * log\, G(z_{c1}, z_{s1}) \qquad (7)$$

Então, para continuar o processo, o modelo recebe o mesmo espaço de conteúdo latente z_{c1} como entrada, mas um espaço latente de estilo gerado aleatoriamente $\hat{y}$ no mesmo formato *hot*. Ambos são concatenados e passados pelo G, gerando uma representação $\tilde{x}$ da frase.

Em seguida, o resultado do discriminador fornece informações para a criação da função de perda para o gerador e codificador L_{adv}, bem como para sua própria evolução de aprendizado L_{dis}. O discriminador é incentivado ao longo do treinamento para resultar em valores mais altos para amostras que ele considera reais, e valores mais baixos para amostras que ele considera falsas ou geradas, como visto na equação 8.

A função de perda que interfere no aprendizado do gerador e codificador L_{adv} é definida pelo log negativo do resultado do discriminador nos dados gerados. Como é treinado para minimização, isso força o gerador e o codificador a gerarem amostras que enganam o discriminador para dar notas mais altas.

$$L_{adv} = \quad -\ log\, D(\tilde{x}, \hat{y}) \quad (8)$$

Da mesma forma, a função de perda que interfere no aprendizado do discriminador L_{dis} é definida pelo resultado negativo da soma do log do resultado dos dados reais com o log de 1 menos o resultado dos dados falsos. Esta função é treinada para minimização, o que obriga o discriminador a tentar maximizar a diferença da pontuação dada aos dados reais e gerados, além de atribuir valores maiores aos dados reais do que aos gerados, como visto na equação 9.

$$L_{dis} = \; - \; (log\; D(x, y) \; + \; (log\; 1 \; - \; log\; D(\tilde{x}, \hat{y}))\;) \quad (9)$$

Essa representação $\tilde{x}$ passa novamente pelo D, que gera os espaços latentes z_{c2} e z_{s2}. Desta vez, o espaço de estilo latente é encorajado a recriar o estilo $\hat{y}$ fornecido a G anteriormente, como visto na equação 10.

$$L_{s2} = \; - \sum \hat{y} \; * \; log\; \bar{z}_{s2} \; (10)$$

A unificação da função de perda de estilo termina pela soma de L_{s1} e L_{s2}, como visto na equação 11.

$$L_s = \; L_{s1} + \; L_{s2} \quad (11)$$

Por fim, o segundo espaço latente de conteúdo gerado é concatenado com o vetor de estilo original da frase, sendo então fornecido a G uma última vez. A partir do resultado desta operação, obtém-se uma representação *hot* da sentença $\tilde{x}$, que é comparada com a sentença original x. Esse processo leva à perda da reconstrução L_{rec}, que ensina o modelo a, principalmente, persistir o espaço latente dos conteúdos z_{c1} e z_{c2} durante a aprendizagem, como visto na equação 12. Esta função é análoga à perda de ciclo.

$$L_{rec} = \; - \sum x \; * \; log\; G(z_{c2}, y) \; (12)$$

5.2.4 Treinamento

Durante o treinamento, o algoritmo é feito seguindo um esquema de treinamento em duas etapas (LAI et al., 2019), a primeira ensinando-o a aprender a reconstrução L_{rec} e identidade, posteriormente adicionando as funções anexadas ao discriminador. O cálculo dos gradientes para minimização ao longo do treinamento é dividido para o gerador, codificador e discriminador, uma vez que as variáveis de cada um desses modelos devem ser otimizadas de acordo com diferentes métricas. Ambas as partes utilizam o otimizador Adam para realização do treinamento.

Na primeira etapa, a função de perda do codificador é descrita pela soma da função de perda de estilo L_s, reconstrução L_{rec} e identidade L_{id}. No caso do gerador, apenas as funções de perda de reconstrução L_{rec} e identidade L_{id} são levadas em consideração, como visto na equação 13.

$$\begin{aligned} E &= min(L_s + L_{rec} + L_{id}) \\ G &= min(L_{rec} + L_{id}) \end{aligned} \qquad (13)$$

No segundo estágio, a função de perda do codificador e do gerador são as mesmas do primeiro estágio, com a adição da função L_{adv}. No caso do discriminador, ele é guiado apenas por L_{dis}, como visto na equação 14.

$$\begin{aligned} E &= min(L_s + L_{rec} + L_{id} + L_{adv}) \\ G &= min(L_{rec} + L_{id} + L_{adv}) \\ D &= min(L_{dis}) \end{aligned} \qquad (14)$$

Por fim, levando em consideração o objetivo do trabalho, o método descrito por este capítulo foi inteiramente pensado para melhorar a retenção do conteúdo na tarefa de transferência de estilo. As duas principais adições foram a separação dos espaços latentes de estilo e conteúdo, e o treinamento supervisionado do espaço de estilo. Tais adições foram capazes de manter um nível de precisão comparável a trabalhos anteriores no que diz respeito a precisão das transferências de estilo, com uma melhoria substancial na preservação do conteúdo. Os resultados práticos que sustentam esta afirmação podem ser analisados no capítulo seguinte.

6 Resultados

Neste capítulo são descritos os resultados obtidos nesta pesquisa. Estes resultados têm como principal objetivo demonstrar a viabilidade do método proposto.

Os principais resultados obtidos permeiam os objetivos deste trabalho, no que diz respeito ao desenvolvimento de um método, baseado na arquitetura CycleGan, capaz de gerar textos melhorando a métrica de preservação de conteúdo em relação a trabalhos prévios.

Um dos problemas atrelados ao objetivo geral do trabalho, é o de possuir uma base de dados não pareados que possuam múltiplos estilos. As bases de dados utilizadas por este trabalho são da Yelp e da Amazon, e tratam de contextos diferentes.

A primeira base de dados foi obtida por meio da tradicional base de inversão de sentimentos da Yelp. Foi utilizada uma abordagem semelhante para rotular frases com sentimento e tempo passado/presente de (LAI et al., 2019). Se uma frase contém pelo menos um verbo no pretérito, a frase foi rotulada como ‘passado’; caso contrário, como 'presente'. O conjunto de dados de produtos da Amazon (RUINING; MCAULEY; 2017) consiste em análises de produtos associados às classificações dos mesmos, em que foram selecionados 4 tipos de produtos: (livros/filmes/eletrônicos/CDs). Além disso, na base de dados da Amazon foi levado em consideração também a inversão de sentimento, totalizando 8 possíveis estilos.

Em relação a base de Yelp, esta é composta de 869.930 frases e 4 rótulos. A Tabela 3 expõe as diferenças no número de exemplos de cada rótulos.

Tabela 3. Estatísticas da base de dados Yelp.

Agrupamento	**Atributo**	**Número de instâncias**
Past	Negativo	219.252
	Positivo	238.560
Present	Negativo	200.552
	Positivo	211.566
Total		**869.930**

Em relação a base da Amazon, esta é composta de 439.811 frases e 8 combinações de atributos. A Tabela 4 expõe as diferenças no número de exemplos de cada atributo.

Tabela 4. Estatísticas da base de dados Amazon.

Agrupamento	**Atributo**	**Número de instâncias**
Livro	Negativo	64.424
	Positivo	73.113
Filme	Negativo	54.169
	Positivo	64.175
Eletronico	Negativo	49.658
	Positivo	57.880
Cd	Negativo	32.596
	Positivo	43.796
Total		**439.811**

Ainda em relação a base de dados utilizada, é importante realçar que as diferenças no comprimento das frases que compõem cada estilo, é uma característica interessante para o problema de gerar frases com gêneros diferentes, visto que o comprimento da frase pode ser aprendido implicitamente pelo algoritmo como característicos de um estilo.

Em função do ambiente computacional disponível, técnicas de pré-processamento como transformação de vetores para *one-hot* foram aplicadas com limite de tamanho de vocabulário de 8.000 palavras, e limite de tamanho de frase com 10 palavras. Em relação a este processo, selecionou-se as 8.000 palavras que mais apareceram nos textos para criar um índice, sendo o vocabulário total conhecido pelo modelo. Todas as frases foram limitadas ou preenchidas para o comprimento de 10 palavras, sendo o preenchimento ou substituição de palavras

desconhecidas realizado pelo termo “[UNK]”, e as palavras conhecidas substituídas pelo índice do vocabulário preestabelecido.

Ainda em relação ao trabalho de pré-processamento destes dados, como mencionado no capítulo do método proposto, a entrada do modelo é um vetor bidimensional esparso, chamado de *one-hot*, que representa estes índices. Sendo as frases compostas por 10 palavras, o vetor de entrada pode ser representado assim como na Figura 18.

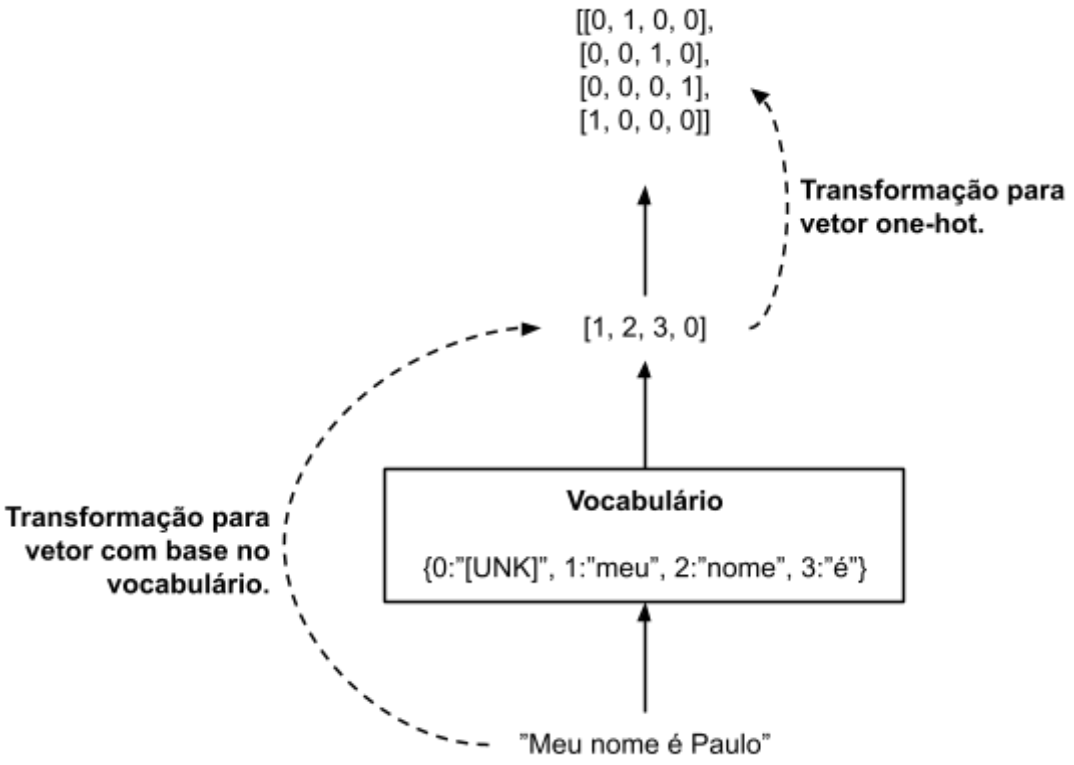

Figura 18. Exemplo do processo de vetorização das frases.

6.1 Treinamento

Uma das principais adições deste trabalho ocorreu pela separação do processo de treinamento em duas etapas, a primeira visou balizar o modelo no que diz respeito a reconstrução e classificação supervisionada do estilo da frase. O acompanhamento visual da mesma pode ser acompanhado na Figura 19.

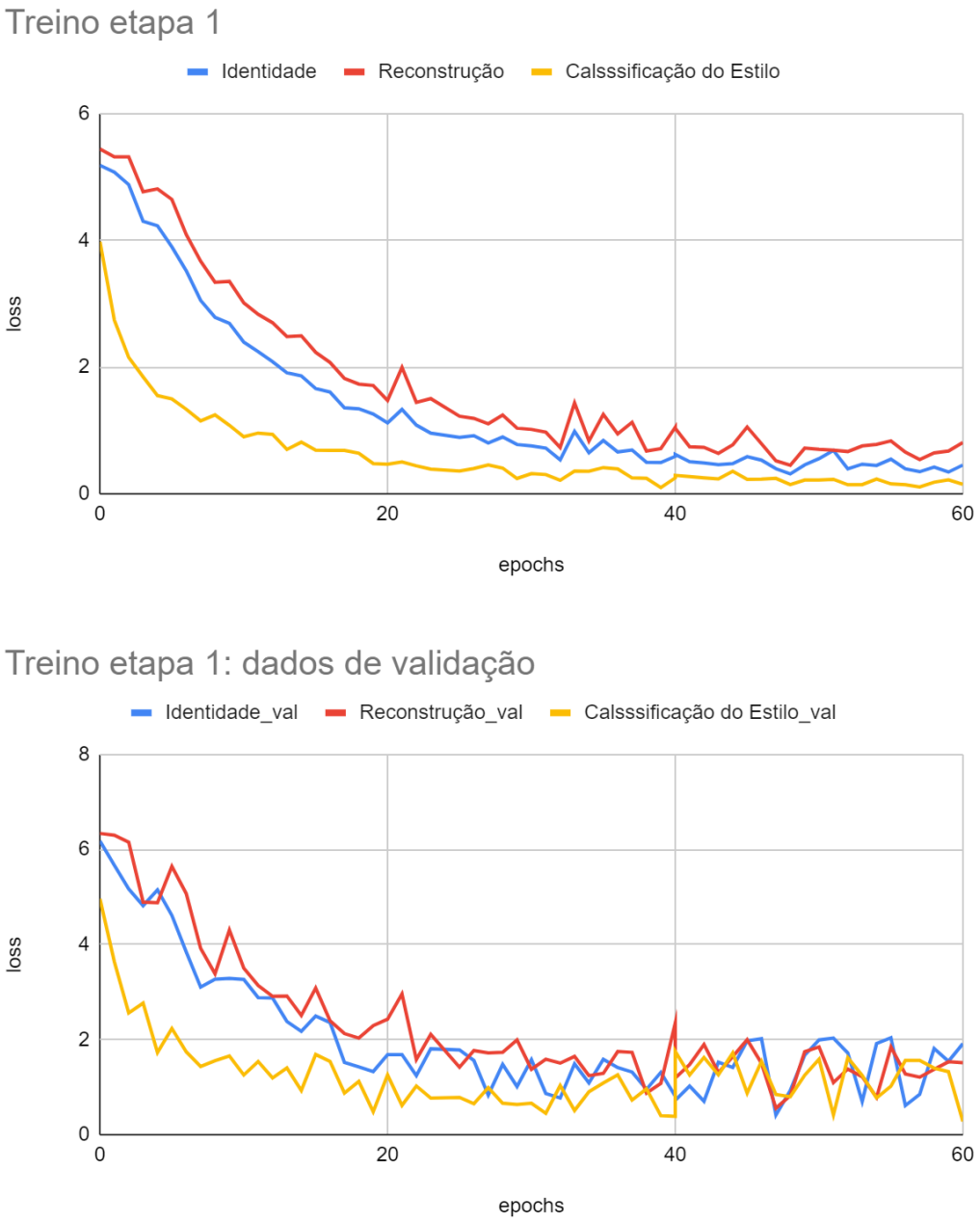

Figura 19. Treinamento do algoritmo CycleGan: etapa 1.

É possível perceber que tanto os resultados de evolução dos dados de treino como o conjunto de validação possuem um comportamento semelhante, reiterando a consistência no aprendizado do algoritmo e a estabilidade ao longo do treino. Uma curiosidade percebida, é a de que já nesta etapa o algoritmo apresenta alguma habilidade de transferência de estilo, antes mesmo da camada adversária da rede. Esta informação é interessante, pois sugere que, eventualmente, processos de treinamento mais simples, mas bem estruturados, podem chegar próximos do

resultado final esperado, sem a necessidade da adição de etapas mais densas e instáveis de treinamento.

Na sequência foi realizada a segunda etapa de treinamento, que leva em consideração as perdas advindas do discriminador, portanto acrescentando as perdas do discriminador e adversária. Nesta etapa foram utilizados dos otimizadores ao longo do processo, para garantir que somente as variáveis corretas estariam sendo atualizadas. O acompanhamento visual da mesma pode ser visto nas seguintes figuras: Figura 20 e Figura 21.

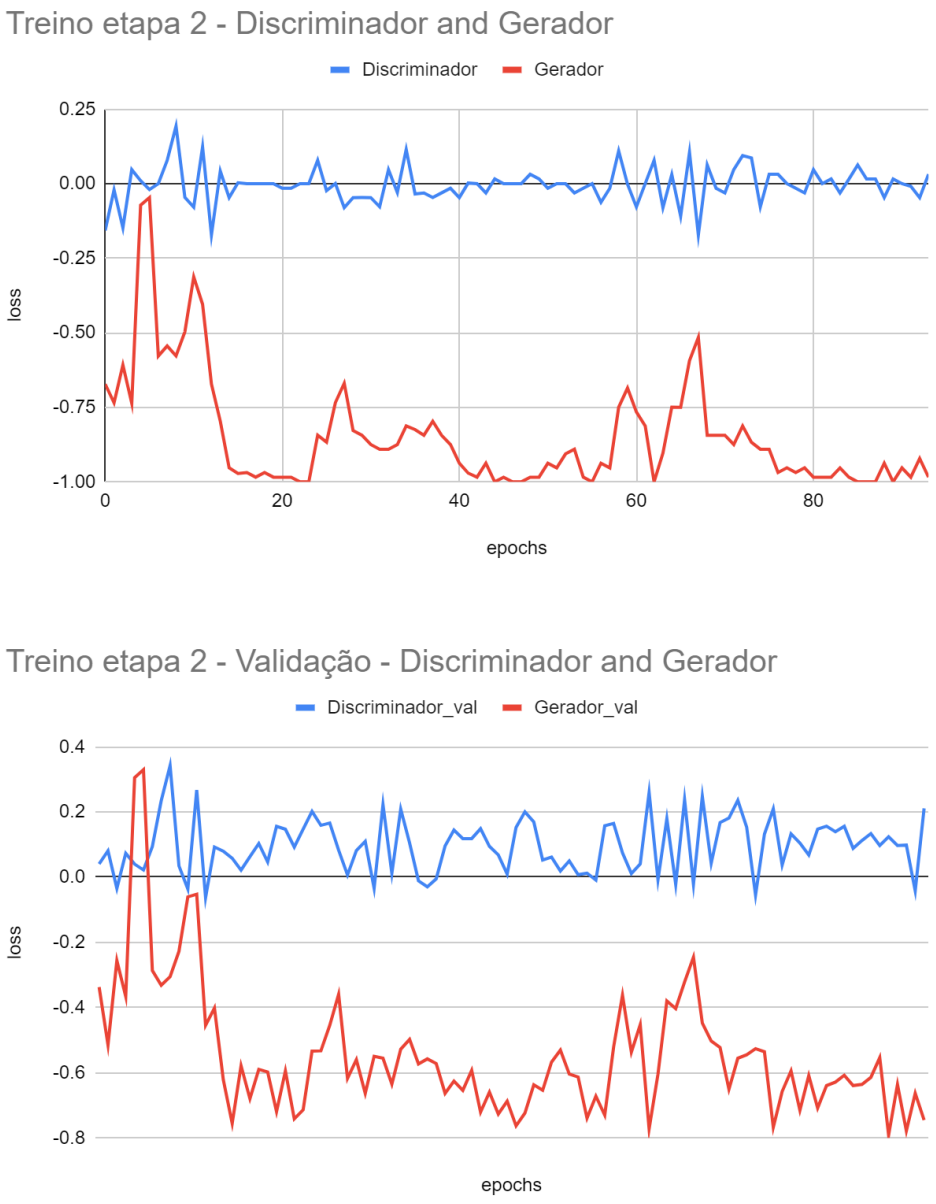

Figura 20. Treinamento do algoritmo CycleGan - Discriminador e Gerador: etapa 2.

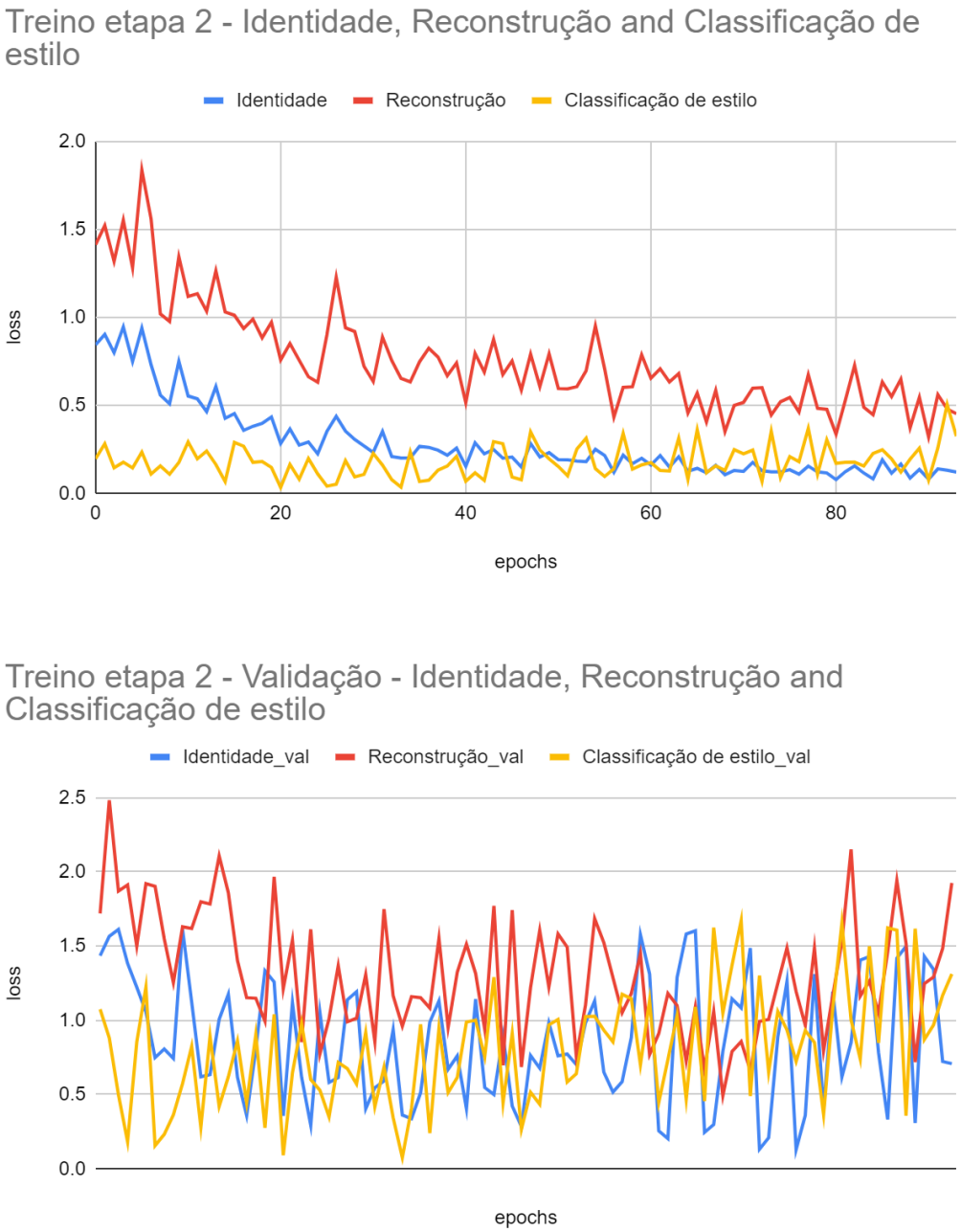

Figura 21. Treinamento do algoritmo CycleGan - Validação, Identidade e Reconstrução: etapa 2.

Percebe-se por meio destas figuras que o treinamento com acréscimo da parte adversária foi mais instável, porém manteve razoável consistência entre os dados de treino e validação. Tal fenômeno pode ser explicado visto que, diferentemente da primeira parte, em que as funções de perda são constantes, resultando em gradientes mais estáveis, nesta segunda etapa ambos os algoritmos estão evoluindo, resultando em funções de perda e gradientes que oscilam.

Além disso, como já explorado por trabalhos prévios, diferente do treinamento padrão de outros algoritmos de rede neural, onde espera-se a minimização literal das funções de perda para se constatar um aprendizado suficiente, no caso das GANs, a incapacidade de continuar evoluindo ou de estabilização é considerado suficiente, pois significa que o discriminador está incapaz de distinguir entre os dados gerados e os reais. Pode-se perceber este comportamento na Figura 20, em que logo no começo o gerador tem um pico em sua perda, mas logo na sequência mantém uma relativa estabilidade na minimização. Este fato pode ter ocorrido pois logo no início quando o discriminador e o gerador ainda não foram treinados, é mais fácil para o discriminador distinguir entre as frases do que para o gerador criar frases convincentes.

6.2 Métrica de precisão de conteúdo

As duas principais métricas utilizadas para medir a performance do algoritmo foram as métricas de precisão na transferência do estilo pretendido e preservação de conteúdo via a métrica BLEU. Para a precisão na transferência do estilo pretendido utilizou-se a entropia cruzada do estilo ideal a ser previsto e a previsão por si só, assim como aplicado por (LAI *et al.*, 2019). Para o cálculo da retenção de conteúdo utilizou-se a métrica BLEU, a fim de medir a similaridade da frase original e da nova pela comparação de n-gramas.

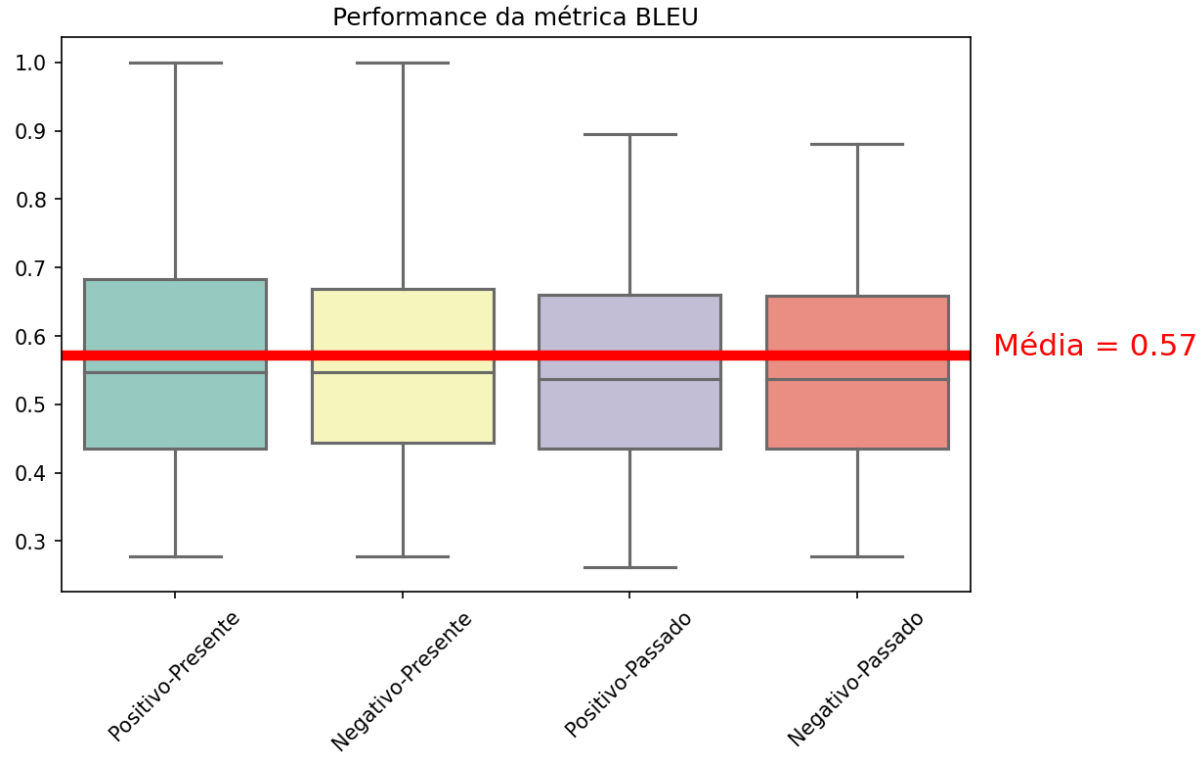

Figura 22. YELP - Métrica BLEU para cada gênero de escrita.

O resultado médio da métrica BLEU na base de dados do Yelp foi de 57.2%. Na Figura 22 é interessante perceber que os tempos verbais no presente possuem uma acurácia superior à do passado. Porém, no geral, todas as combinações possuem uma acurácia semelhante, o que mostra que apesar de existir um sutil desbalanceamento na base de dados, esta não foi significativa para os resultados finais.

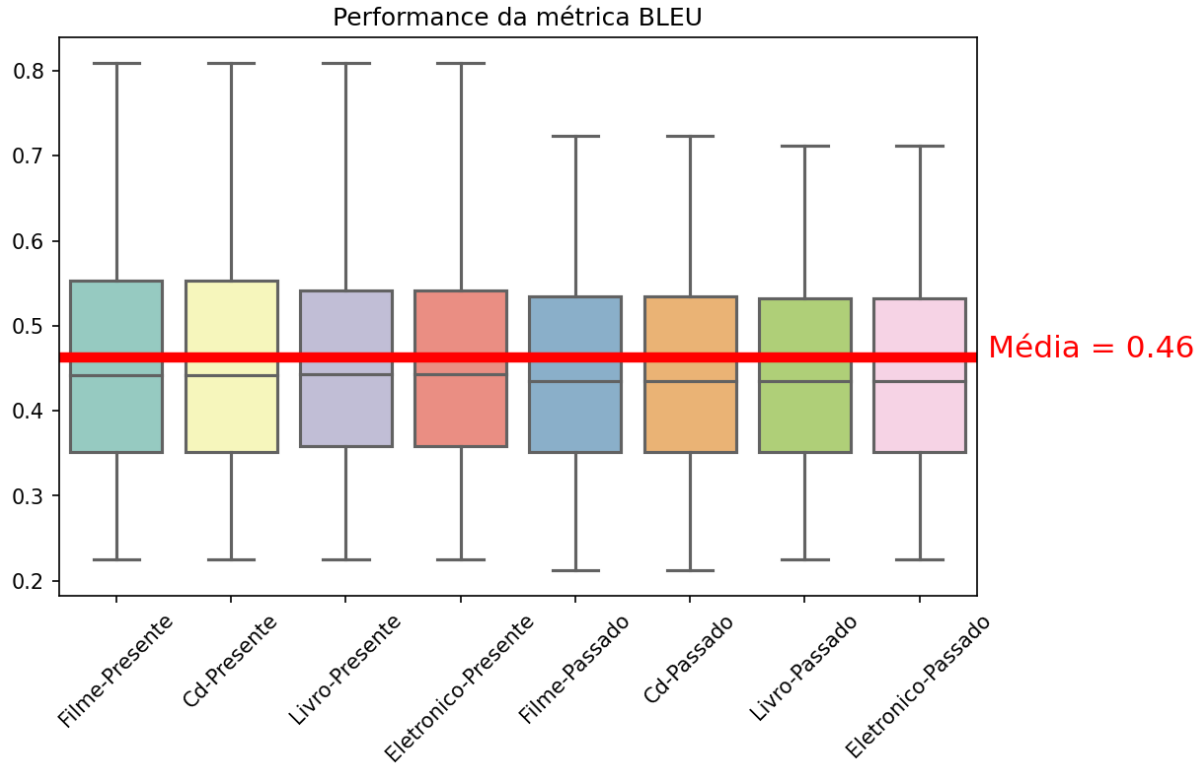

Figura 23. AMAZON - Métrica BLEU para cada gênero de escrita.

O resultado médio da métrica BLEU na base de dados do Yelp foi de 46.2%. Em relação a Figura 23, seguindo o mesmo padrão da Figura 19, os tempos verbais no presente possuem uma acurácia superior aos no passado, demonstrando que este estilo possui de fato uma complexidade maior que o presente.

Será abordado ainda neste capítulo a comparação com o estado da arte, porém, os resultados apresentados acima são superiores, indicando que as modificações feitas impactaram de forma satisfatória a métrica de precisão.

6.3 Comparação com trabalhos prévios

Em relação à comparação dos resultados obtidos com o estado da arte, o método proposto se propõe a maximizar a métrica BLEU, alcançando um valor mais próximo de 0.5 ou 50%. Neste sentido, com a análise das seguintes tabelas, verifica-se que este objetivo foi atingido, tendo também obtido uma performance comparável nas métricas de precisão de estilos, tanto no dataset da Yelp com o

sentimento e tempo verbal, quanto no dataset da Amazon com o sentimento e o tipo de produto.

Tabela 5. YELP - Comparação dos resultados com trabalhos anteriores.

Yelp - sent/tense			
	Style acc	Tense acc	BLEU
(Subrama-nian et al., 2018)	0.75	0.91	25.9%
(Logeswaran et al., 2018)	0.76	0.94	14.7%
(LAI, C. T. et al., 2019)	**0.79**	**0.96**	32.2%
Nosso modelo	0.75	0.89	**57.2%**

Tabela 6. AMAZON - Comparação dos resultados com trabalhos anteriores.

Amazon - sent/prod			
	Style acc	Prod acc	BLEU
(Subrama-nian et al., 2018)	**0.79**	**0.9**	15.3%
(Logeswaran et al., 2018)	0.75	0.81	11.2%
(LAI, C. T. et al., 2019)	0.76	0.87	22.4%
Nosso modelo	0.76	0.85	**46.2%**

Percebe-se a partir dos resultados da Tabela 5 e 6 que o método proposto por este trabalho tem potencial pois, apesar de resultar em uma performance inferior ou igual nas métricas de estilo, nosso trabalho melhorou consideravelmente uma métrica que ainda estava muito aquém do sugerido e esperado pela literatura. Tal melhoria se deu pelas adições feitas tanto na arquitetura do modelo, quanto no processo de treinamento, sendo o foco deste trabalho.

6.4 Exemplo de frases geradas

A fim de expor alguns resultados práticos, a Tabela 5 apresenta as transferências entre algumas frases da base de dados YELP. É importante ressaltar que as frases estão em inglês por conta da base de dados utilizadas, porém, estas poderiam ser traduzidas, ou o modelo inteiro poderia ser treinado novamente em uma base de dados análoga em português.

Tabela 7.YELP - Comparação das frases geradas por este trabalho em relação aos trabalhos prévios.

YelpTense: (past, negative) → (present, positive)	
Original	i think that's really ridiculous especially when their service sucked .
Nosso modelo	i think that's really great especially when their service was outstanding.
(LAI, C. T. et al., 2019)	i think it's really great especially when their service rocks .
(Logeswaran et al., 2018)	i think that's really worth the time when their service is awesome .
(Subrama-nian et al., 2018)	i really love how much .
YelpTense: (past, positive) → (present, negative)	
Original	they were extremely professional and clean with the repairs .
Nosso modelo	they are extremely unprofessional and dirty with the repairs .
(LAI, C. T. et al., 2019)	they are extremely unprofessional and dirty with the repairs .
(Logeswaran et al., 2018)	they are extremely unprofessional and not professional with the repairs
(Subrama-nian et al., 2018)	they are extremely unprofessional and not .

É possível perceber que o modelo proposto possui resultados similares aos de (LAI, C. T. et al., 2019), por ter sido inspirado em um formato de arquitetura semelhante. Porém, em alguns exemplos é notória a melhor retenção de conteúdo entre a frase original e os métodos testados anteriormente.

Os resultados obtidos na Tabela 7 demonstram um comportamento semelhante, confirmando que a retenção de conteúdo supera os trabalhos prévios, e se compara no quesito de precisão na transferência de estilo.

Tabela 8. AMAZON - Comparação das frases geradas por este trabalho e trabalhos prévios.

AmaProd: (book, negative) → (movie, positive)	
Original	i really do feel i wasted my time on this egotistical book .
Nosso modelo	i really do feel great the time spent on this modest movie.
(LAI, C. T. et al., 2019)	i really did i look my time for this dvd christmas movie .
(Logeswaran et al., 2018)	i really really really enjoy my time on this movie .
(Subrama-nian et al., 2018)	i really i have this movie in my car .
AmaProd: (electronic, positive) → (CD, negative)	
Original	i purchased this mount a few months ago .
Nosso modelo	i hate this cd a few months ago .
(LAI, C. T. et al., 2019)	i hate this cd a few months ago .
(Logeswaran et al., 2018)	i bought this album a few years ago .
(Subrama-nian et al., 2018)	this cd sucks .

De forma geral os resultados obtidos com este capítulo estão de acordo com as expectativas e sustentam que as mudanças de arquitetura propostas podem ser um caminho interessante na criação de algoritmos que alterem menos as sentenças depois de uma transferência de estilo.

7 Conclusão

Esta dissertação propõe um método para realizar a transferência de estilo com conjuntos de dados não paralelos entre vários estilos. Propomos um modelo de sequência para sequência, baseado na arquitetura CycleGan, com separação dos espaços latentes de estilo e conteúdo, aprendizado supervisionado do espaço de estilo e treinamento seguindo um esquema de duas etapas. Os resultados experimentais demonstram que nosso modelo supera consideravelmente os trabalhos anteriores em relação à preservação de conteúdo e mantém a precisão comparável em relação a transferência de estilo. Isso significa, na prática, que nosso modelo pode transferir as frases para o estilo pretendido alterando menos as frases, que é o comportamento ideal e esperado de um método como esse.

Em se tratando das hipóteses levantadas no início do estudo:

- H1) A separação dos espaços latentes de estilo e conteúdo contribui para a melhoria da métrica BLEU.
- H2) O treinamento supervisionado do espaço latente de estilo contribui para a melhoria da métrica BLEU.
    - Ambas as hipóteses 1 e 2 trataram da pertinência e resultados práticos das modificações feitas ao longo da arquitetura e no modo de treinamento. Ambas as hipóteses puderam ser constatadas tanto pelos resultados individuais, quanto pela comparação frente aos trabalhos anteriores.
- H3) É possível melhorar a métrica BLEU sem impactar na acurácia na métrica de precisão de estilo
    - Sim. No capítulo de resultados pode-se observar que este trabalho resultou em uma acurácia comparável nas métricas de estilo, melhorando a preservação de conteúdo.

Embora nosso modelo alcance melhor preservação do conteúdo, a precisão do estilo transferido pode ser considerada uma limitação, que pode ser melhorada no futuro utilizando arquiteturas com camadas de atenção, entre outras técnicas disponíveis na literatura.

Além disso, outra limitação do projeto são as métricas utilizadas para medir a precisão dos modelos. Como já destacado em trabalhos anteriores, projetar métricas de avaliação adequadas ainda é um problema em aberto para a transferência de estilo de texto. Embora as métricas BLEU sejam precisas, o ideal para um trabalho como esse seria realizar testes com avaliadores humanos, já que a métrica BLEU não considera a complexidade do problema em sua totalidade. O desenvolvimento de uma métrica que vá além de uma comparação de n-gramas e se aproxime de uma avaliação humana pode agregar muito na avaliação de modelos futuros.

Para projetos futuros, além da preservação de conteúdo e precisão da conversão de estilo, novos conjuntos de dados podem ser testados e criados para medir o desempenho de método semelhantes a esses em cenários mais complexos. Tal análise é importante porque, como visto nos resultados deste trabalho e dos anteriores, no dataset da Amazon, por exemplo, estilos que possuem campos semânticos mais ricos exigem mais dos modelos para serem igualmente precisos. Tarefas que exploram cenários que vão além das conversões de frases e tempos verbais podem sugerir novas técnicas de balanceamento de dados, aumento de dados ou mesmo método.

Referências

ARJOVSKY, M.; BOTTOU, L. Towards Principled Methods for Training Generative Adversarial Networks. 17 jan. 2017. Disponível em: http://arxiv.org/abs/1701.04862. Acesso em: 17 mar. 2021.

ARJOVSKY, M.; CHINTALA, S.; BOTTOU, L. Wasserstein GAN. 26 jan. 2017. Disponível em: http://arxiv.org/abs/1701.07875. Acesso em: 6 abr. 2021.

BAHDANAU, D.; CHO, K.; BENGIO, Y. Neural Machine Translation by Jointly Learning to Align and Translate. 1 set. 2014. Disponível em: http://arxiv.org/abs/1409.0473. Acesso em: 12 mar. 2021.

CHEN, J.; YIN, D.; HUANG, S.; DAI, X. Utilizing Non-Parallel Text for Style Transfer by Making Partial Comparisons. [*s. l.*], p. 5379–5386, 2019.

CHEN, L.; DAI, S.; TAO, C.; SHEN, D.; GAN, Z.; ZHANG, H,; ZHANG, Y.; CARIN, L. Adversarial Text Generation via Feature-Mover's Distance. *In*: 32ND CONFERENCE ON NEURAL INFORMATION PROCESSING SYSTEMS, 2020. Anais [...]. [*S. l.: s. n.*], 2020. p. 125–135.

CHEN, X.; DUAN, Y.; HOUTHOOFT, R.; SCHULMAN, J.; SUTSKEVER, I.; ABBEL, P. InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. 12 jun. 2016. Disponível em: http://arxiv.org/abs/1606.03657. Acesso em: 12 mar. 2021.

CHEN, X. *et al.* Towards unsupervised text multi-style transfer with parameter-sharing scheme. Neurocomputing, [*s. l.*], v. 426, p. 227–234, 2021.

CHO, K. *et al.* Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. [*S. l.: s. n.*], 2014. Disponível em: https://doi.org/10.3115/v1/d14-1179

DARANI, L. H. Persuasive Style and its Realization Through Transitivity Analysis: A SFL Perspective. [*S. l.: s. n.*], 2014. Disponível em: https://doi.org/10.1016/j.sbspro.2014.12.066

FU, Z. *et al.* Style Transfer in Text: Exploration and Evaluation. 18 nov. 2017. Disponível em: http://arxiv.org/abs/1711.06861. Acesso em: 6 abr. 2021.

GATT, A.; KRAHMER, E. Survey of the State of the Art in Natural Language Generation: Core tasks, applications and evaluation. [*S. l.: s. n.*], 2018. Disponível em: https://doi.org/10.1613/jair.5477

GATYS, L.; ECKER, A.; BETHGE, M. A Neural Algorithm of Artistic Style. [*S. l.: s. n.*], 2016. Disponível em: https://doi.org/10.1167/16.12.326

GOODFELLOW, I. J. *et al.* Generative Adversarial Networks. 10 jun. 2014. Disponível em: http://arxiv.org/abs/1406.2661. Acesso em: 17 mar. 2021.

HAN, M.; WU, O.; NIU, Z. Unsupervised Automatic Text Style Transfer Using LSTM. [*S. l.: s. n.*], 2018. Disponível em: https://doi.org/10.1007/978-3-319-73618-1_24

HU, M.; HE, M. Non-parallel text style transfer with domain adaptation and an attention model. Applied Intelligence, [*s. l.*], 2021. Disponível em: https://doi.org/10.1007/s10489-020-02077-5

HU, Z.; LEE, R. K.-W.; AGGARWAL, C. C. Text Style Transfer: A Review and Experiment Evaluation. 24 out. 2020. Disponível em: http://arxiv.org/abs/2010.12742. Acesso em: 8 fev. 2021.

HE, J.; WANG, X.; NEUBIG, G.; BERG, T. Toward Controlled Generation of Text. 2 mar. 2017. Disponível em: http://arxiv.org/abs/1703.00955. Acesso em: 5 abr. 2021.

IDA AYU PUTU; LEE, J.-H. Sequence to Sequence CycleGAN for Non-Parallel Sentiment Transfer with Identity Loss Pretraining. *In*: PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON RESEARCH IN ADAPTIVE AND

CONVERGENT SYSTEMS, 2020. Anais [...]. [*S. l.: s. n.*], 2020. Disponível em: https://doi.org/10.1145/3400286.3418249

JAIN, P.; MISHRA, A.; AZAD, A.; SANKARANARAYANAN, K. Unsupervised controllable text formalization, Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, p. 6554–6561, 2019.

JANG, E.; GU, S.; POOLE, B. Categorical Reparameterization with Gumbel-Softmax. 3 nov. 2016. Disponível em: http://arxiv.org/abs/1611.01144. Acesso em: 6 abr. 2021.

JIN, D.; JIN, Z.; ZHOU, J.; ORRI, L. SZOLOVITS, P. Hooks in the Headline: Learning to Generate Headlines with Controlled Styles. [*S. l.: s. n.*], 2020. Disponível em: https://doi.org/10.18653/v1/2020.acl-main.456

JING, Y.; YANG, Y.; FENG, Z.; YE, J.; YU, Y.; SONG, M. Neural Style Transfer: A Review. IEEE transactions on visualization and computer graphics, [*s. l.*], v. 26, n. 11, p. 3365–3385, 2020.

JOHN, V.; MOU, L.; BAHULEYAN, H.; VECHTOMOVA, O. Disentangled representation learning for non-parallel text style transfer. *In*: PROCEEDINGS OF THE 57TH ANNUAL MEETING OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS, 2019, Florence, Italy. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Stroudsburg, PA, USA: Association for Computational Linguistics, 2019. Disponível em: https://doi.org/10.18653/v1/p19-1041

JOHNSTONE, B. Linguistic strategies and cultural styles for persuasive discourse. [*S. l.: s. n.*], 1989.

KAPTEIN, M.; MARKOPOULOS, P.; RUYTER, B.; AARTS, E. Personalizing persuasive technologies: Explicit and implicit personalization using persuasion profiles. [*S. l.: s. n.*], 2015. Disponível em: https://doi.org/10.1016/j.ijhcs.2015.01.004

KAWASHIMA, T.; TAKAGI, T. Sentence Simplification from Non-Parallel Corpus with

Adversarial Learning. *In*: INTERNATIONAL CONFERENCE ON WEB INTELLIGENCE, 2019. Anais [...]. [*S. l.: s. n.*], 2019.

KINGMA, D. P.; WELLING, M. Auto-Encoding Variational Bayes. 20 dez. 2013. Disponível em: http://arxiv.org/abs/1312.6114. Acesso em: 15 abr. 2021.

LAI, C. T.; HONG, Y. T.; CHEN, H. Y.; LU, C. J.; LIN, S. D. Multiple Text Style Transfer by using Word-level Conditional Generative Adversarial Network with Two-Phase Training. *In*: PROCEEDINGS OF THE 2019 CONFERENCE ON EMPIRICAL METHODS IN NATURAL LANGUAGE PROCESSING, 2019. Anais [...]. [*S. l.: s. n.*], 2019.

LIN, K.; LIU, M. Y.; SUN, M. T.; KAUTZ, J. Learning to generate multiple style transfer outputs for an input sentence, Proceedings of the Fourth Workshop on Neural Generation and Translation, pp. 10–23, 2020.

LOGESWARAN, L.; LEE, H.; BENGIO, S. Content preserving text generation with attribute controls. Advances in Neural Information Processing Systems, 5108–5118, 2018.

LI, D.; ZHANG, Y.; GAN, Z.; CHENG, Y.; BROCKETT, C.; SUN, M. T.; DOLAN, B. Domain Adaptive Text Style Transfer. [*S. l.: s. n.*], 2019. Disponível em: https://doi.org/10.18653/v1/d19-1325

METZ, L.; POOLE, B.; PFAU, D.; DICKSTEIN, J. S. Unrolled Generative Adversarial Networks. 7 nov. 2016. Disponível em: http://arxiv.org/abs/1611.02163. Acesso em: 17 mar. 2021.

MIRZA, M.; OSINDERO, S. Conditional Generative Adversarial Nets. 6 nov. 2014. Disponível em: http://arxiv.org/abs/1411.1784. Acesso em: 12 abr. 2021.

MUEHLENHAUS, I. If Looks Could Kill: The Impact of Different Rhetorical Styles on Persuasive Geocommunication. [*S. l.: s. n.*], 2012. Disponível em: https://doi.org/10.1179/1743277412y.0000000032

PAPINENI, K; ROUKOS, S.; WARD, T.; ZHU, W. J. BLEU: a method for automatic evaluation of machine translation. *In*: THE 40TH ANNUAL MEETING, 2002/7/7-2002/7/12, Philadelphia, Pennsylvania. Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02. Morristown, NJ, USA: Association for Computational Linguistics, 2001. Disponível em: https://doi.org/10.3115/1073083.1073135

PENG, G. F.; YANG, Y. S.; TSAI, C. Y.; SOO, V. W. Generate Modern Chinese Poems from News Based on Text Style Transfer Using GAN. [*S. l.: s. n.*], 2019. Disponível em: https://doi.org/10.1109/taai48200.2019.8959907

PIETRA, D. A Statistical Approach to Machine Translation. Computational linguistics, [*s. l.*], 1990.

RUINING , H.; MCAULEY, J. Modeling the visual evolution of fashion trends with one-class collaborative filtering. International World Wide Web Conferences Steering Committee., [S. l.], p. 507–517, 2022. Disponível em: http://dx.doi.org/10.1145/2872427.2883037.

SANTOS, C. N.; MELNYK, I.; PADHI, I. Fighting Offensive Language on Social Media with Unsupervised Text Style Transfer. [*S. l.: s. n.*], 2018. Disponível em: https://doi.org/10.18653/v1/p18-2031

SCHMIDT, R.; BRAUN, S. Generative Text Style Transfer for Improved Language Sophistication, 2020. Disponível em: http://cs230.stanford.edu/projects_winter\ _2020/reports/32069807.pdf. Acesso em: 19 jan 2022.

SHEN, T.; LEI, T.; BARZILAY, R.; JAAKKOLA, T. Style Transfer from Non-Parallel Text by Cross-Alignment. *In*: 31ST CONFERENCE ON NEURAL INFORMATION PROCESSING SYSTEMS, 2017. Anais [...]. [*S. l.: s. n.*], 2017. Disponível em: https://doi.org/10.1007/s10489-020-02077-5

BALTRUŠAITIS, T.; AHUJA, C.; MORENCY, L. P. Multimodal Machine Learning: A

Survey and Taxonomy. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019.

HUANG, Y.; ZHU, W.; XIONG, D.; ZHANG, Y.; HU, C.; XU, F. Cycle-consistent adversarial autoencoders for unsupervised text style transfer. In :Proceedings of the 28th International Conference on Computational Linguistics, 2020.

XU, J.; SUN, X.; ZENG, Q.; REN, X.; ZHANG, X.; WANG, H.; LI, W. Unpaired Sentiment-to-Sentiment Translation: A Cycled Reinforcement Learning Approach. 14 maio 2018. Disponível em: http://arxiv.org/abs/1805.05181. Acesso em: 16 abr. 2021.

ZHU, J. Y.; PARK, T.; ISOLA, P.; EFOS, A. A. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. 2017 IEEE International Conference on Computer Vision (ICCV), [*s. l.*], 2017. Disponível em: http://doi.org/10.1109/ICCV.2017.244