UNIVERSIDADE FEDERAL FLUMINENSE

ARTHUR MARIANO ROCHA DE AZEVEDO SCALERCIO

Transferência de Estilo Textual Não-Supervisionada com Modelos de Linguagem Mascarados

NITERÓI

2021

UNIVERSIDADE FEDERAL FLUMINENSE

ARTHUR MARIANO ROCHA DE AZEVEDO SCALERCIO

Transferência de Estilo Textual Não-Supervisionada com Modelos de Linguagem Mascarados

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Mestre em Computação. Área de concentração: Ciência da Computação

Orientadora:
Aline Marins Paes Carvalho

NITERÓI

2021

Ficha catalográfica automática - SDC/BEE
Gerada com informações fornecidas pelo autor

S279t Scalercio, Arthur Mariano Rocha de Azevedo
Transferência de Estilo Textual Não-Supervisionada com Modelos de Linguagem Mascarados / Arthur Mariano Rocha de Azevedo Scalercio ; Aline Marins Paes Carvalho, orientador. Niterói, 2021.
97 f. : il.

Dissertação (mestrado)-Universidade Federal Fluminense, Niterói, 2021.

DOI: http://dx.doi.org/10.22409/PGC.2021.m.74202251253

1. Processamento de linguagem natural (Computac?o). 2. Aprendizado de maquina. 3. Produção intelectual. I. Carvalho, Aline Marins Paes, orientador. II. Universidade Federal Fluminense. Instituto de Computação. III. Título.

CDD -

Bibliotecário responsável: Debora do Nascimento - CRB7/6368

ARTHUR MARIANO ROCHA DE AZEVEDO SCALERCIO

Transferência de Estilo Textual Não-Supervisionada com Modelos de Linguagem Mascarados

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Mestre em Computação. Área de concentração: Ciência da Computação.

Aprovada em setembro de 2021.

BANCA EXAMINADORA

Profa. Dra. Aline Marins Paes Carvalho - Orientadora, UFF

Prof. Dr. Leandro Augusto Frata Fernandes, UFF

Profa. Dra. Maria José Bocorny Finatto, UFRGS

Profa. Dra. Viviane Pereira Moreira, UFRGS

Niterói

2021

Dedico essa dissertação à minha mãe, Neide, e à minha mulher, Bianca, as quais sempre me incentivaram incondicionalmente. Também dedico ao meu avô, Mariano (in memorian), por todo o carinho que me deu. Forçoso dedicar essa dissertação também a todos os professores que tive até hoje e que me habilitaram para essa conquista.

Agradecimentos

Agradeço ao apoio incondicional dado por todos os membros da minha família nessa jornada e à minha orientadora Professora Aline pela suavidade e qualidade na condução de nossos estudos. Também agradeço à banca pela avaliação e à Universidade Federal Fluminense por oferecer o curso.

Resumo

A tarefa de transferência de estilo textual (TST) tem como objetivo alterar de forma automática traços estilísticos de um texto, como formalidade, sentimento, estilo autoral, humor, complexidade, entre outros, mas tentando garantir a preservação de seu conteúdo. Embora essa tarefa seja investigada desde a década de 80, recentemente ela tem ganhado destaque na área de processamento de língua natural (PLN) com aprendizado de máquina, mais especificamente na sub-área de geração de textos em língua natural, devido ao desenvolvimento de estratégias não-supervisionadas para contornar a ausência de dados paralelos para treinamento. Por outro lado, os modelos de linguagem induzidos a partir de uma grande quantidade de dados auto-anotados têm impulsionado o desenvolvimento de diversas abordagens para resolver tarefas da área de processamento de língua natural que vão além da geração de texto, incluindo modelos de linguagem mascarados e autorregressivos. Entretanto, enquanto o uso de modelos de linguagem mascarados pré-treinados alcançou resultados do estado da arte em diversas tarefas de PLN, eles ainda são pouco explorados na tarefa de transferência de estilo. Esta dissertação explora o uso de um modelo de linguagem mascarado pré-treinado como um dos componentes de um método não-supervisionado para abordar a tarefa de transferência de estilo. Dado que modelos de linguagem mascarados apresentam dificuldades para a tarefa de geração de texto, devido a sua natureza não autorregressiva, nesta dissertação o modelo de linguagem mascarado não é usado de forma direta na tarefa de transferência de estilo, mas para transferir o conhecimento contido nele para um modelo convencional de sequência-para-sequência (*Seq2Seq*), que executará, de fato, a transferência de estilo. A partir das representações expressivas obtidas pelo modelo de linguagem pré-treinado baseado na arquitetura *Transformer*, espera-se que a qualidade dos textos gerados seja aprimorada. A proposta foi avaliada em duas tarefas de transferência de estilo: imitação autoral e transferência de sentimento. Bases de dados vastamente usadas para essas duas tarefas foram utilizadas para realizar avaliações quantitativas, qualitativas e com pessoas nos resultados do modelo proposto e comparar com os resultados obtidos em métodos que obtiveram resultados no estado da arte. Em ambas as tarefas, a abordagem proposta superou os outros métodos na métrica de controle de estilo, enquanto obteve resultados competitivos na métrica de preservação de conteúdo.

Palavras-chave: Transferência de Estilo Textual; Transformers; Modelo de Linguagem Mascarado; Redes Neurais Profundas; Geração de Linguagem Natural; Destilação de Conhecimento; Albert; Aprendizado Não-Supervisionado.

Abstract

Text style transfer (TST) aims to automatically change a text's stylistic features, such as formality, sentiment, authorial style, humor, complexity, among others, while still trying to preserve its content. Although this task has been investigated since the 1980s, it has only recently gained more attention in natural language processing with machine learning, more specifically in the natural language generation sub-area. Such attention is due primarily to the development of unsupervised strategies to circumvent the absence of parallel data for supervised training. On the other hand, language models induced from a large amount of self-annotated data have driven the development of several approaches to solve natural language processing tasks that go beyond text generation, including masked language models and autoregressive. However, while the use of pre-trained masked language models has achieved state-of-the-art results in several tasks involving attempts to understand language, they are still underexplored in the style transfer task. This dissertation explores using a pre-trained masked language model as one of the components of an unsupervised method to address the style transfer task. Given that masked language models present difficulties for the text generation task, due to their non-autoregressive nature, in this dissertation, the masked language model is not used directly in the style transfer task but to transfer the knowledge contained in it to a conventional sequence-to-sequence (*Seq2Seq*) model, which is the main component to perform the style transfer. From the expressive representations obtained by the pre-trained language model based on the *Transformer* architecture, it is expected that the quality of the generated texts will be improved. The proposal was evaluated in two style transfer tasks: authorial imitation and feeling transfer. Widely used databases for these two tasks were used to carry out quantitative, qualitative, and human evaluations on the results of the proposed model and compare with the results obtained in methods that obtained state-of-the-art results. The proposed approach outperformed the other methods in the style control metric in both tasks while achieving competitive results in the content preservation metric.

Keywords: Text Style Transfer; Transformers; Masked Language Model; Deep Neural Networks; Natural Language Generation; Knowledge Distillation; Albert; Unsupervised Learning.

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

PLN	:	Processamento de Língua Natural;
NLG	:	Geração de Textos em Língua Natural;
LM	:	Modelo de Linguagem Autorregressivo;
MLM	:	Modelo de Linguagem Mascarado;
DLSM	:	Deep Latent Sequence Model;
IA	:	Inteligência Artificial;
TLCE	:	Termo de Consentimento Livre e Esclarecido;

Sumário

Capítulo 1

Introdução

A área de Processamento de Língua Natural (PLN) [64, 45, 70] é uma área de conhecimento na interseção da Linguística Computacional [28] e da Inteligência Artificial (IA) [71] que tem como objetivos a modelagem e a construção de modelos computacionais que permitam a interação entre humanos e computadores por meio da língua que é natural aos seres humanos. O grau de dificuldade para a realização destas tarefas varia, indo desde processos mais simples, como checagem de grafia [97], até tarefas mais complexas, como responder perguntas automaticamente que demandariam compreensão de textos [82]. Desde seus primórdios, a área de PLN tem abordado a resolução de tarefas com o auxílio de métodos de aprendizado de máquina e estatística [7]. Essa parceria tem sido ainda mais alavancada recentemente, com o ressurgimento dos métodos baseados em redes neurais desenvolvidos na área de aprendizado profundo [25, 95, 59], que têm alcançado resultados no estado da arte em tarefas nas mais diversas áreas do conhecimento humano [55, 17]. Apesar dos resultados expressivos e de alguns trabalhos afirmarem que estão avançando no desenvolvimento de métodos computacionais que permitam que computadores *entendam* e *interpretem* a língua natural, alguns pesquisadores argumentam que há um mal entendimento do relacionamento entre forma linguística e significado e que, da maneira como os modelos neurais são treinados, eles não aprendem significado [3]. De todo modo, a área tem visto resultados expressivos em *benchmarks* diversos de PLN.

Como dito anteriormente, diversas tarefas de PLN têm sido abordadas recentemente a partir do desenvolvimento de métodos da área de aprendizado profundo. Em geral, na área de PLN, tais métodos são baseados na construção e utilização de modelos de linguagem neurais pré-treinados [4, 95, 75, 17, 8, 20]. Os modelos de linguagem neurais tanto se baseiam na formulação auto-regressiva clássica, em que a palavra seguinte é predita a

partir das palavras que apareceram antes, de forma unidirecional [8], como também em proposições recentes de modelos de linguagem mascarados [17], em que uma palavra em qualquer posição é predita a partir das demais palavras no contexto, bidirecionalmente. Em ambos os casos, a construção de tais modelos requer um volume considerável de textos de exemplos, que, para o pré-treinamento do modelo de linguagem, são auto-anotados a partir do próprio conteúdo textual. Uma prática que tem se tornado comum é iniciar o aprendizado de uma tarefa a partir desses modelos pré-treinados e extrair representações ou ajustar os pesos da rede destes modelos para aprender a resolver a tarefa final [39, 84, 76]. Entretanto, para resolver tarefas finais, tais como classificação, resposta a consultas, sumarização, entre outras, em geral, ainda são necessários exemplos anotados, ou seja, exemplos que representem a entrada e a saída esperada de um modelo que resolva a tarefa.

Dentre as tarefas definidas como complexas, a *Geração de Textos em Língua Natural* (NLG, do inglês *Natural Language Generation*) [98] é uma sub-área de PLN que tem feito vasto uso de métodos de aprendizado de máquina [88, 41, 80, 8], dando origens a diversas aplicações que incluem tarefas como sumarização de textos [85], simplificação textual [92] e geração automática de histórias [21], entre outras. Certamente, a geração de textos demanda compreensão e interpretação, de forma a produzir textos que sejam semanticamente corretos, fluentes, e coesos. As habilidades de compreensão e geração de textos são características inerentemente humanas e dotar máquinas com esse comportamento, se concretizável, será um avanço considerável para a área de Inteligência Artificial.

No contexto de tarefas que geram texto, esta dissertação tem como foco a tarefa de *Transferência de Estilo Textual* (TST) [66, 38, 53, 31, 88, 56], que consiste em alterar propriedades estilísticas de um texto, por exemplo, formalidade, sentimento, humor, complexidade, estilo autoral, mas mantendo o contexto semântico original. Imbuir métodos computacionais de tal habilidade tem o potencial de criar aplicações capazes de transformar um texto escrito de maneira informal em um texto formal, ou converter um texto de um tom agressivo para um tom mais gentil. Na indústria, por exemplo, já existem ferramentas de auxílio de escrita que fazem uso de algoritmos de aprendizado de máquina voltados à tarefa de transferência de estilo. Há ferramentas[1] que permitem, por exemplo, que uma pessoa troque o estilo de um texto para deixá-lo mais claro, mantendo o conteúdo da mensagem. A utilização da transferência de estilo permeia não somente o processo de transformação textual, mas também a avaliação do teor estilístico de textos [74]. Por exemplo, uma empresa pode recomendar que um colaborador seja mais formal

[1] https://hemingwayapp.com/

em seus emails de trabalho, a partir de uma análise de sua escrita em conteúdos formais. Outra aplicação é na área de desenvolvimento de robôs que literalmente conversam com seres humanos. Em [47] foram conduzidos experimentos para analisar o impacto do estilo de escrita dos robôs nas ações do usuário. Descobriu-se que quando um estilo de conversa mais casual é utilizado, participantes tem menos chances de serem persuadidos a tomar uma ação quando comparados com participantes que conversavam com um robô usando um estilo de conversa formal. Métodos de transferência de estilo também podem ser usados para gerar paráfrases de autores [108]. Assim, da mesma forma que outras técnicas de IA estão sujeitas ao mau uso, é preciso avaliar o impacto ético de técnicas de transferência de estilo. Como evitar, por exemplo, o uso de tais técnicas para geração de plágio autoral ou até mesmo que alguém se passe por outra pessoa? Por outro lado, o próprio uso dessas técnicas para complementar e enrobustecer os atuais modelos de detecção de plágio pode ser uma resposta a esse problema.

Considerando as aplicações promissoras que podem surgir, a tarefa de transferência de estilo textual tem um histórico longo dentro da área de PLN e recentemente tem recebido notória visibilidade principalmente devido aos resultados promissores obtidos com modelos de redes neurais profundas [66, 9, 108, 88, 31, 53, 15]. As aplicações práticas são inúmeras, visto ser recorrente em nossa sociedade a necessidade de adaptar textos a situações diferentes, de acordo com a audiência ou objetivos desejados, mas com ajustes que não impliquem em alteração no conteúdo semântico original.

Abordar essa tarefa com aprendizado de máquina é desafiador, particularmente por duas razões: a primeira diz respeito à dificuldade de avaliar automaticamente a qualidade de um texto gerado por um método computacional, herdada da tarefa base de geração de texto. Apesar de haver métricas automáticas de avaliação, cada uma delas se preocupa com determinada dimensão a ser medida, de modo que a combinação dessas métricas apenas fornece uma avaliação superficial da qualidade do texto gerado. Por exemplo, quando se utiliza a acurácia de um classificador pré-treinado como métrica, a preocupação principal é somente se o texto gerado é parecido com textos no estilo desejado. Essa métrica sozinha não indica se a transferência de estilo de fato foi satisfatória, pois o texto pode até parecer com o estilo desejado, mas o conteúdo pode ser totalmente diferente. O segundo desafio reside no fato de ser difícil obter pares de sentença que possuam o mesmo conteúdo e estilos diferentes, o que dificulta bastante o problema de treinar um modelo de aprendizado de máquina para transferir estilo.

Assim, embora a atual capacidade de transferir e transformar estilos a partir de textos

possa ser uma boa medida do avanço da área de Inteligência Artificial, o progresso na transferência de estilo de textos ainda não apresenta resultados tão significativos como em alguns outros domínios. Essas dificuldades são oriundas dos desafios citados anteriormente, como a ausência de dados de entrada e saída pareados e a ausência de métricas de avaliação mais robustas. Outro complicador é a natureza discreta de textos, tornando mais difícil a modelagem quando comparada com domínios contínuos, como é o caso das imagens. Assim, apesar de os modelos e as técnicas de aprendizado profundo terem contribuído também para a melhora do desempenho da tarefa de transferência de estilo textual, dada sua complexidade, ainda há bastante espaço para melhorias. O presente estudo introduz técnicas que almejam a qualidade da geração de textos dentro da tarefa de transferência de estilo, vislumbrando aproximar o texto gerado por máquinas dos textos gerados por seres humanos.

1.1 Problema de Pesquisa

Antes de adentrar nos objetivos específicos que o presente estudo busca alcançar, convém fazer um breve apanhado de alguns conceitos, para uma melhor compreensão da proposta desta dissertação, como segue.

1. *LINGUAGEM* é "um conjunto complexo de processos - resultado de uma certa atividade psíquica profundamente determinada pela vida social - que torna possível a aquisição e o emprego de uma *Língua* qualquer"[89]. Usa-se também o termo para designar todo sistema de sinais que serve de meio de comunicação entre os indivíduos. Desde que se atribua valor convencional a determinado sinal, existe uma *LINGUAGEM*. [13]

2. *LÍNGUA* é um sistema gramatical pertencente a um grupo de indivíduos. Expressão da consciência de uma coletividade, a *LÍNGUA* é o meio pelo qual ela concebe o mundo que a cerca e sobre ele age. A utilização social da faculdade da linguagem, não é imutável; ao contrário, tem de viver em perpétua evolução, paralela ao organismo social que a criou [13].

3. *DISCURSO* é a língua no ato, na execução individual. E, como cada indivíduo tem em si um ideal linguístico, ele procura extrair do idioma de que se serve as formas de enunciado que melhor exprimam o gosto e o pensamento. Essa escolha entre os diversos meios de expressão que lhe oferece o rico repertório de possibilidades, que é a língua, denominamos *ESTILO* [13].

Essas três denominações aplicam-se a aspectos diferentes, mas não opostos, do fenômeno extremamente complexo que é a comunicação humana. Assim, considerando o estilo como sendo "o aspecto e a qualidade que resultam da escolha entre os meios de expressão disponíveis" [65], a presente dissertação propõe um modelo que converte textos de um estilo de origem s para um outro estilo alvo desejado s'. Considerando a dificuldade na obtenção de dados paralelos, esse estudo focou em um modelagem não-supervisionada, onde o conjunto de exemplos é composto somente de sentenças textuais e o seu respectivo estilo, ou seja, assume-se que não existe disponibilidade de sentenças pareadas para o treinamento.

Nesta dissertação, tenta-se representar o estilo de uma maneira ampla, podendo ser o estilo de escrita de um indivíduo, por exemplo Machado de Assis, até diferentes variações internas da língua reconhecidas pela literatura. Como exemplos dessas diferenças internas da língua, elencam-se 1) as diferenças no espaço geográfico, ou variações diatópicas (falares locais, variantes regionais e até intercontinentais); 2) diferenças entre as camadas socioculturais, ou variações diastráticas (nível culto, língua padrão, nível popular, etc) e 3) diferenças entre os tipos de modalidade expressiva, ou variações diafásicas (língua falada, língua escrita, língua literária, linguagem especiais, linguagem dos homens, linguagem das mulheres, etc.). Assim, para atender a essa amplitude de possibilidades de estilo, o estilo é abstraído como um atributo de aprendizado, de forma que o método proposto seja capaz de aprender a resolver não apenas a transferência de um único estilo rígido, mas de mais de um estilo.

Como veremos no Capítulo 2, os trabalhos anteriores na literatura desenvolvem métodos para a tarefa de transferência de estilo não-supervisionada com redes neurais de duas maneiras: (1) considerando métodos de edição de sentenças, os quais buscam encontrar os trechos da sequência que indicam o estilo e substituí-los por trechos no estilo desejado [56, 105]; e (2) considerando métodos que recebem como entrada uma sequência e aprendem a gerar a sequência inteira de saída, conhecidos como métodos sequência-para-sequência (*seq2seq*) [93],[15]. Os métodos que editam a sentença de entrada para gerar a sentença de saída funcionam bem somente para algumas tarefas, uma vez que, na maioria das tarefas, não é possível assumir que a transferência de estilo será realizada pela simples substituição de um trecho da sentença por outro. Na tarefa de imitação autoral, por exemplo, é comum se alterar completamente o texto original, tornando inviável o uso de métodos de edição de sentença. Já os modelos sequência-para-sequência conseguem ser mais genéricos e capazes de englobar diversas sub-tarefas de transferência de estilo. Por outro lado, embora alguns métodos comportem modelos de linguagem mascarados, nenhum deles procurou extrair o conhecimento de um modelo mascarado e tirar vantagem

das suas ricas representações bidirecionais, que é a proposta desta dissertação, conforme discutido a seguir.

1.2 Objetivos

No contexto da tarefa de transferência de estilo de texto, a presente dissertação almeja alcançar três objetivos. O primeiro é propor um método baseado em aprendizado não-supervisionado com redes neurais que se beneficie de modelos de linguagem mascarados pré-treinados. O segundo objetivo é treinar um modelo a partir desse método e avaliar seus resultados, de forma que ele que tenha um desempenho ao menos competitivo com modelos que obtiveram resultados no estado da arte. O terceiro objetivo é ter um modelo que seja genérico o suficiente para lidar com mais de um tarefa de transferência de estilo.

O uso de modelos de linguagem mascarados pré-treinados baseados na arquitetura Transformer, como BERT [17], tem alcançado resultados impressionantes em várias tarefas de PLN. Diversas pesquisas são motivadas pelo fato que tais modelos são pré-treinados e disponibilizados de forma gratuita, para sua reutilização ou refinamento. Além disso, os modelos de linguagem mascarados conseguem resolver tarefas que vão além da geração de textos, ao capturar os aspectos sintáticos e semânticos necessários para predizer a palavra mascarada [83]. Por outro lado, o uso de modelos de linguagem mascarados em tarefas de geração de texto é menos popular. Isso é muito devido ao fato de a arquitetura de modelos como BERT [17] ser composta somente de um codificador, enquanto tarefas de geração de linguagem natural normalmente são implementadas usando uma arquitetura codificador-decodificador, que se foca em produzir uma palavra y_i, dada uma sentença de entrada X, e as palavras que vieram antes de y_i $(y_1, \ldots, y_{i-1})$, a partir da distribuição de probabilidade $P(y_i|y_1, \ldots, y_{i-1}, X)$. Essa dissertação procura mostrar que técnicas de mascaramento, incluindo o uso de modelos de linguagem mascarados, também podem ser úteis em modelos de geração de linguagem utilizados para resolver a tarefa de transferência de estilo.

Para confirmar a hipótese de que a tarefa de transferência de estilo pode se beneficiar de modelos de linguagem mascarados, foram escolhidas as tarefas de imitação autoral [108] e de transferência de sentimento [56], ambas em inglês, dada a disponibilidade de *benchmarks* publicados abertamente e de forma livre e avaliados na literatura recente. Apesar de a língua escolhida durante os experimentos ter sido a inglesa, a aplicação do método em outras línguas é imediata, dada a existência e curadoria de sentenças em cada estilo

para o treinamento do modelo. A primeira tarefa investigada nessa dissertação consiste em parafrasear uma sentença conforme o estilo de um autor. A segunda consiste em trocar o sentimento de um texto de positivo para negativo e vice-versa, preservando o resto do conteúdo semântico. Para facilitar o entendimento do que se deseja alcançar, alguns exemplos de sentenças e suas traduções desejadas, para a tarefa de imitação autoral, constam na Tabela 1.1 e, para a tarefa de transferência de sentimento, constam na Tabela 1.2.

Tabela 1.1: Sentenças e suas traduções na tarefa de imitação autoral

Estilo	Sentença
Inglês Moderno	Have you spoken to him?
Inglês Literário	Hast thou met with him?
Inglês Moderno	Goodbye.
Inglês Literário	Farewell.
Inglês Moderno	No, Romeo will respond to the letter's writer, telling him whether he accepts the challenge.
Inglês Literário	Nay, he will answer the letter's master, how he dares, being dared.

Tabela 1.2: Sentenças e suas traduções na tarefa de transferência de sentimento

Estilo	Sentença
Sentimento Positivo	ever since joes has changed hands it's gotten better and better.
Sentimento Negativo	ever since joes has changed hands it's just gotten worse and worse.
Sentimento Positivo	definitely not disappointed that i could use my birthday gift!
Sentimento Negativo	definitely disappointed that i could not use my birthday gift!
Sentimento Positivo	blue cheese dressing was above average.
Sentimento Negativo	blue cheese dressing wasn't the best by any means.

Considerando, de um modo geral, que obras literárias não possuem traduções escritas em outros estilos ou não possuem dados traduzidos em quantidadade para treinar modelos, e que o processo de anotação para criação de sentenças paralelas inter-domínios é demorado e custoso, esta dissertação se valeu de uma abordagem não-supervisionada para treinamento do modelo para ambas as tarefas-alvo, onde somente dados brutos em dois domínios distintos são usados durante o treinamento. Como se adotou uma abordagem não-supervisionada durante treinamento, onde não se tem acesso a dados paralelos, mais importante se torna a necessidade de um volume mínimo de sentenças para obtenção de um modelo que generalize bem.

A maioria dos trabalhos em transferência de estilo de texto usam uma abordagem onde um codificador busca obter uma representação oculta do conteúdo do texto que não dependa do estilo, enquanto que o decodificador busca gerar um texto de mesmo conteúdo com o estilo desejado [88]. O método proposto em [53] mostrou que, além de ser difícil obter representações do conteúdo desacopladas do estilo, tal desacoplamento

não é necessário na tarefa de transferência de estilo. Nessa linha, o método proposto nesta dissertação também parte da hipótese que o modelo não precisa separar conteúdo e estilo e, consequentemente, não assume a existência de variáveis ocultas que representem o conteúdo da sentença.

Em termos de avaliação da tarefa, objetiva-se que o texto gerado pelo modelo: (1) tenha o estilo desejado; (2) preserve o conteúdo semântico da sequência de entrada; e (3) seja fluente. Métricas distintas foram usadas para medir cada um desses atributos. Para verificar se o texto foi gerado conforme o estilo desejado, mensuramos a acurácia de um classificador que foi pré-treinado usando textos de ambos os estilos. Para medir a preservação de conteúdo, característica mais importante que um modelo de transferência de estilo deveria possuir, três métricas foram usadas: BLEU [72], BertScore [113] e similaridade semântica (SIM) [99]. Finalmente, para mensurar a fluência do texto gerado, foi mensurada a perplexidade que um modelo de linguagem pré-treinado no respectivo domínio estilístico atribuirá à sentença gerada. Como cada métrica citada mede determinado atributo da sequência gerada, os textos gerados foram submetidos à avaliação com pessoas para uma avaliação global. Os resultados mostraram que o uso de um modelo de linguagem mascarado pré-treinado ao longo do treinamento principal do modelo melhora a qualidade dos textos gerados, alcançando melhor resultados nas três dimensões analisadas nas avaliações automáticas, assim como nas avaliações humanas. Dessa forma, a grande vantagem do modelo proposto, quando comparado a um modelo que não faz uso de uma técnica de destilação de conhecimento, reside no fato de o método proposto conseguir controlar melhor o estilo do texto gerado, mantendo a métrica de preservação de conteúdo no mesmo patamar.

1.2.1 Questões de Pesquisa

Para atingir os objetivos propostos, a investigação levará em consideração as seguintes perguntas de pesquisa:

Q1: *Como modelos de linguagem mascarados podem ser utilizados para melhorar o desempenho de modelos neurais na tarefa de transferência de estilo e consequentemente gerar textos de alta qualidade?*

Q2: *Como contornar as dificuldades que modelos de linguagem mascarados enfrentam para gerar textos quando eles são inseridos no processo de transferência de estilo?*

Q3: *Como um modelo de TST que se vale de um modelo de linguagem mascarado se*

compara a outros modelos no estado da arte?

Como solução, incluiu-se em um modelo neural, que executa o processo de transferência de estilo, um componente que faz uso de técnicas de mascaramento. O método proposto foi nomeado de MATTES[2], do inglês *MAsked Transformer for TExt Style transfer*. Para avaliá-lo e compará-lo a outros trabalhos relacionados, adotou-se uma metodologia experimental e os resultados obtidos, de acordo com ele, foram mensurados.

Nessa dissertação, a inclusão de modelos mascarados para a tarefa de transferência de estilo foi inspirada por um método de mascaramento que se vale de modelos de linguagem mascarados, como BERT [17] e ALBERT [54], para melhorar a qualidade dos textos gerados, e consequentemente melhorar o desempenho da tarefa de transferência de estilo. Os resultados experimentais indicaram que a extração de conhecimento através do uso de um modelo de linguagem mascarado pré-treinado é benéfica para melhorar o desempenho do modelo.

1.3 Metodologia

Inicialmente, realizou-se uma revisão da literatura referente à tarefa de transferência de estilo textual, englobando as principais arquiteturas e técnicas de aprendizado de máquina usadas. A metodologia para construção do modelo segue [25]. Construiu-se um modelo probabilístico que faz uso de redes neurais, determinou-se uma função de custo a ser minimizada pelo modelo e as métricas de avaliação para o conjunto de validação e de testes foram escolhidas. Após isso, repetidamente, foram realizados experimentos, para ajustes de hiperparâmetros e promovendo ajustes de código, quando necessário. O melhor modelo no conjunto de validação, de acordo com critérios definidos, foi escolhido para gerar resultados no conjunto de testes. Para melhor compreensão dos experimentos, o fluxo de vida de um (1) experimento é mostrado na Figura 1.1. Como se observa, os únicos insumos de um experimento são o conjunto de dados e um ALBERT pré-treinado.

Os resultados obtidos foram avaliados quantitativamente usando-se cinco métricas automáticas de avaliação: acurácia (Acc) dada por um classificador neural pré-treinado para avaliar o controle de estilo do texto gerado; BLEU [72], BertScore [113] e similaridade semântica (SIM) [99] para avaliar como o modelo preserva o conteúdo da sentença original; e a perplexidade (PPL) de acordo com um modelo de linguagem pré-treinado

[2]Implementação disponível em https://github.com/MeLLL-UFF/unsup-style-transfer-text

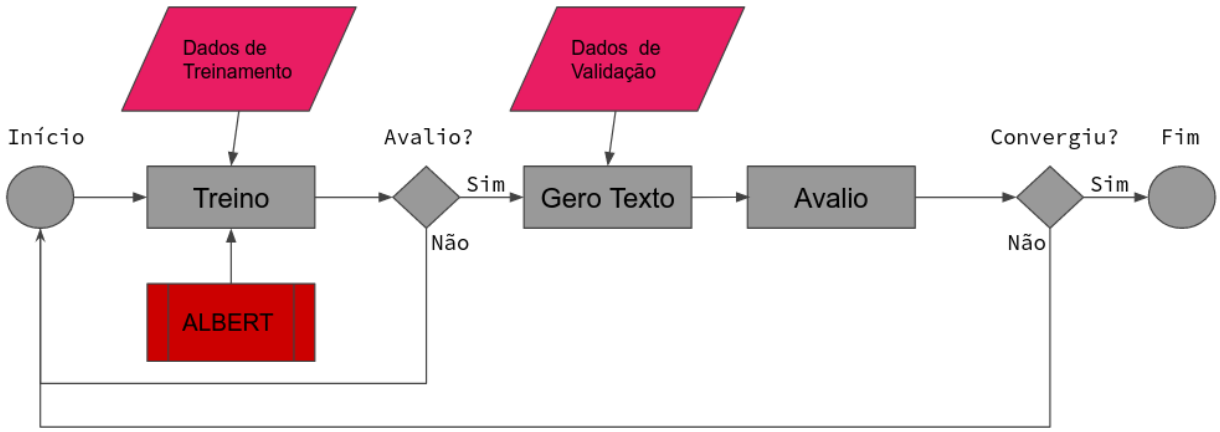

Figura 1.1: Fluxo de vida de um experimento

para mensurar a fluência do texto gerado. Avaliações com pessoas foram conduzidas para analisar qualitativamente os textos produzidos pelo modelo.

A arquitetura usada segue [15]. Como bloco arquitetural principal do modelo, utilizou-se a rede neural Transformers [95], que segue a abordagem sequência para sequência (*Seq2Seq*) amplamente usado em PLN e que tem obtido os resultados do estado da arte em diversas tarefas. Ele é composto por um codificador e um decodificador, os quais são blocos de redes de neurais de atenção interconectados. Além de uma sequência textual, o modelo treinado também precisa de um estilo como variável de entrada. Como saída, o modelo busca gerar uma sequência que tenha o mesmo conteúdo da sequência de entrada, escrita no estilo informado na entrada. Se o estilo de entrada for o estilo da sequência de entrada, o modelo deverá, idealmente, reescrever a entrada.

Na tarefa de imitação autoral, o conjunto de dados usado foi uma coletânea de peças de *shakespeare* traduzidas linha a linha para o inglês moderno. O conjunto de dados foi coletado por [108] e usado em trabalhos anteriores em transferência de estilo textual não supervisionado [31, 50]. Apesar de esse conjunto de dados ser composto de sentenças paralelas nos dois estilos tratados, essa informação não é usada no treinamento. Essas sentenças paralelas são usadas somente para fins de avaliação nos conjuntos de validação e de teste. Com relação à tarefa de transferência de sentimento, foi usado o conjunto de dados YELP, coletado por [88], que são avaliações de estabelecimentos realizadas dentro do aplicativo YELP. Ele contém 250K sentenças negativas e 380K sentenças positivas. Como conjunto de testes, usamos 1000 sentenças paralelas anotadas por pessoas, introduzidas por [56].

1.4 Contribuições

As contribuições trazidas pela presente dissertação são as que seguem:

1. Um novo método de treinamento não-supervisionado, o qual é capaz de *extrair conhecimento* a partir de um modelo de linguagem mascarado pré-treinado.

2. Até onde temos conhecimento, na tarefa de transferência de estilo, esse é o primeiro estudo que usa um *modelo de linguagem mascarado como parte da função de custo.*

3. Nos experimentos, que englobaram as tarefas de imitação autoral e de transferência de sentimento, MATTES superou outras abordagens pretéritas. Especificamente, conseguimos melhorar tanto o controle de estilo do texto gerado quanto a preservação do conteúdo da sentença gerada.

1.5 Organização do Texto

O restante da dissertação está organizado da seguinte forma. O Capítulo 2 traz a explicação teórica das principais técnicas de aprendizado de máquina usadas nessa dissertação, assim como os principais trabalhos relacionados ao tema de transferência de estilo não-supervisionada. No Capítulo 3, o modelo proposto e a sua implementação são descritos. No Capítulo 4, são descritos os experimentos realizados juntamente com os resultados obtidos. Por fim, no último capítulo são relembrados o problema abordado, os resultados que foram alcançados e as limitações da solução proposta. A dissertação encerra-se com prováveis trabalhos futuros e considerações finais.

Capítulo 2

Fundamentação Teórica

Neste capítulo, serão descritos os conceitos e técnicas de aprendizado de máquina necessários ao entendimento da proposta desta dissertação. A técnica de aprendizado de máquina utilizada nesta dissertação tem como base a sub-área de aprendizado de máquina denominada de aprendizado profundo, que, por sua vez, contempla em sua maioria métodos baseados em redes neurais. Assim, esse capítulo inclui uma visão geral sobre aprendizado de máquina e redes neurais na Seção 2.1. Ademais, o método aqui proposto tem como objetivo utilizar aprendizado de máquina e modelos de linguagem para resolver um problema da área de Processamento de Língua Natural (PLN). Então, nas Seções 2.2, 2.3 e 2.4 serão apresentados os princípios e técnicas mais específicos da área de aprendizado de máquina aplicados ao PLN, que foram necessários para alcançar os objetivos propostos. Finalmente, na Seção 2.5, serão abordados os principais trabalhos relacionados ao tema de Transferência de Estilo Textual Não-Supervisionada.

2.1 Aprendizado de Máquina e Redes Neurais

Um algoritmo de aprendizado de máquina é um algoritmo capaz de aprender a partir de experiência. Mitchell [68] nos fornece uma definição do que consiste aprendizado de máquina: *"Diz-se que um programa aprende a partir da experiência E com relação a alguma tarefa T e métrica de desempenho P, se a performance na tarefa T, medida por P, melhora com a experiência E"*. Os tipos de experiência E, tarefas T e métricas de desempenho P são dos mais variados e buscamos apenas fornecer um norte teórico e não definir formalmente o que pode ser usado em cada uma dessas entidades. O maior desafio para os algoritmos de aprendizado de máquina é que eles devem apresentar um bom desempenho em dados de entrada não vistos durante o seu aprendizado, e não ir bem somente nos

dados de entrada vistos durante o treinamento. A habilidade de um modelo ter um bom desempenho em dados até então não vistos é chamada de generalização. Tipicamente, tem-se acesso a um conjunto de dados de treinamento; durante o treinamento computa-se o valor de uma função, em geral baseada no erro de treinamento, chamada de função de perda ou de custo. No entanto, diferente de um problema de otimização, espera-se que o erro de generalização, também conhecido como erro de teste, seja tão baixo quanto o erro de treinamento. O erro de teste corresponde ao valor esperado do erro quando o modelo receber uma entrada não vista anteriormente. Em aprendizado de máquina, o erro de teste é normalmente mensurado em um conjunto de testes que foi coletado separadamente do conjunto de treino. Assim, há dois desafios centrais no aprendizado de máquina. O primeiro é tornar o erro de treinamento pequeno. O segundo é tornar a diferença entre o erro de treinamento e o erro de teste pequeno. No primeiro caso, trata-se de evitar *underfitting* e no segundo caso de evitar *overfitting*. O primeiro ocorre quando não é possível obter um erro baixo no conjunto de treinamento. O segundo ocorre quando a diferença entre o erro de treinamento e o erro de testes é muito grande.

Algoritmos de aprendizado de máquina pode ser divididos em dois grupos. **Algoritmos de aprendizado não-supervisionados** utilizam um conjunto de dados que contêm características de cada amostra e aprendem propriedades da estrutura dos dados. **Algoritmos de aprendizado supervisionados** utilizam um conjuntos de dados que também contêm características, mas cada amostra está associada a um rótulo ou alvo. De um modo geral, algoritmos não-supervisionados envolvem a observação de vários exemplos de um vetor $\mathbf{x}$ e procuram implicitamente ou explicitamente aprender uma distribuição de probabilidade $p(\mathbf{x})$, ou propriedades interessantes dessa distribuição; enquanto algoritmos supervisionados envolvem a observação de vários exemplos de um vetor $\mathbf{x}$ e um valor ou vetor $\mathbf{y}$ associado, para aprender a prever $\mathbf{y}$ a partir de $\mathbf{x}$, normalmente estimando $p(\mathbf{y}|\mathbf{x})$. Quando algoritmos de treinamento não-supervisionado, durante o treinamento, criam alguma espécie de supervisão, eles são chamados de auto-supervisionados.

Redes neurais são o pilar de modelos de aprendizado profundo [25]. O objetivo delas é aproximar uma função f^*. Uma rede neural define um mapeamento $\mathbf{y} = f(\mathbf{x}, \theta)$ e aprende os valores dos parâmetros θ que resultem na melhor aproximação da função. São chamadas de redes porque são representadas por um grafo de computação, que indica como a função f pode ser obtida a partir da composição de várias funções. Por exemplo, podemos ter três funções f^1, f^2, f^3 conectadas em uma cadeia para formar $f(\mathbf{x}) = f^3(f^2(f^1(\mathbf{x})))$. Nesse caso, f^1 é chamada de primeira camada, f^2 é chamada de segunda camada, e assim sucessivamente. O comprimento total da cadeia fornece a profundidade do modelo. O

nome Aprendizado Profundo (em inglês, *Deep Learning*) surgiu dessa terminologia. A última camada da rede neural é chamada de camada de saída e as intermediárias são chamadas de camadas ocultas. Redes Neurais possuem grande capacidade de generalizar funções. O teorema da aproximação universal [37, 14] estabelece que uma rede neural com uma camada de saída linear e pelo menos uma camada oculta que tenha uma função de ativação não-linear pode aproximar qualquer função cujo domínio seja um subconjunto de $\mathbb{R}^n$. No entanto, apesar de, teoricamente, uma rede neural com uma única camada intermediária ser suficiente para representar qualquer função, na prática, essa camada poderá ser incrivelmente grande e pode falhar no aprendizado e na generalização. Em várias circunstâncias, usar redes neurais de várias camadas, ou seja, modelos profundos pode reduzir o número de unidades necessárias para representar a função desejada e pode reduzir o erro de generalização. Apesar de redes neurais com múltiplas camadas terem sido propostas desde a década de 80, a área que passou a ser chamar Aprendizado Profundo ganhou notoriedade quando em [34] foi demonstrado que uma rede neural era capaz de superar o método SVM com núcleo RBF [6] no conhecido *benchmark* MNIST para reconhecimento de imagens. Apesar de outros algoritmos mais simples de aprendizado de máquina terem funcionado bem para muitas tarefas, eles não tiveram sucesso em problemas centrais da Inteligência Artificial cujo espaço de atributos é de alta dimensionalidade, como reconhecimento de discurso, reconhecimento de imagens e processamento de língua natural [25]. Assim, o aprendizado profundo objetiva superar os desafios que surgem para generalizar novos exemplos quando lida-se com dados de grande dimensão, assim como os elevados custos computacionais impostos por esses espaços vetoriais de dimensão elevada. Através da adição de mais camadas e de mais unidades dentro de uma camada, uma rede neural profunda pode representar funções de alto grau de complexidade [25]. Dessa forma, o grande poder de generalização de redes neurais, combinado com o grande avanço dos *frameworks* computacionais de cálculo numérico, foram os fatores dominantes para a explosão do uso de redes neurais profundas, também na área de PLN. Na seção seguinte, será abordada uma arquitetura de rede profunda muita usada quando se lida com texto, conhecidas como modelos Sequência-para-Sequência [93].

2.2 Modelos Sequência-para-Sequência (Seq2Seq)

Modelos Sequência-para-Sequência, *Seq2Seq*, ou *Sequence-to-sequence*, são arcabouços baseados em aprendizado profundo que lidam com uma classe de tarefas de PLN em que se deseja obter, a partir de uma sequência de valores de entrada, uma sequência de valores

como saída, e não somente um valor. Como exemplo dessas tarefas, podem ser citadas:

- Tradução: converter uma sentença ou um texto de uma língua para outra.
- Conversação: receber uma afirmação ou questão e respondê-la.
- Sumarização: receber um texto como entrada e devolver um resumo do texto.

O paradigma Seq2Seq é relativamente novo, com sua primeira aplicação publicada em 2014 [93] para a tarefa de tradução de Inglês-Francês. Em alto nível, um modelo Seq2Seq é composto por duas redes neurais, uma codificadora e outra decodificadora.

Os primeiros modelos usavam redes neurais recorrentes como os componentes codificador e decodificador. Nesses modelos, o codificador recebe uma sequência como entrada e codifica-a em um vetor de contexto de tamanho fixo. Assim, o papel do codificador é ler a sequência de entrada $< x_1, \ldots, x_n >$, onde x_i é um token, e gerar um vetor C de dimensão fixa que represente o contexto. Após isso, o decodificador busca gerar a sequência de saída $< y_1, \ldots, y_m >$, iniciando a partir do vetor de contexto C. Para tanto, o vetor de contexto obtido a partir da entrada é usado para inicializar o estado oculto do decodificador. Em uma rede recorrente do tipo sequência-para-sequência, cada token x_i da sequência de entrada gera uma representação oculta h_i e normalmente o vetor de contexto é o vetor do estado oculto do último token da sentença de entrada. A Figura 2.1 ilustra o funcionamento de uma rede recorrente Seq2Seq na tarefa de tradução automática do Inglês para o Português.

Para gerar o primeiro token de saída y_1, o decodificador usa normalmente um token especial que indica o início do processo de geração (por exemplo [CLS]), junto com o vetor de contexto que funciona como estado oculto inicial da rede decodificadora h_0. Os próximos tokens a serem gerados seguem o mesmo procedimento. O decodificador usa o token gerado anteriormente em adição ao estado oculto atual, de forma que $y_t = f(y_{t-1}, h_t)$. O processo de geração continua até que surja um token que indique o término da geração.

O grande problema no uso de redes recorrentes como componente base de modelos Seq2Seq está no fato de ser muito difícil comprimir uma sequência de tamanho variável em um único vetor de contexto, principalmente quando a sequência de entrada for longa. Assim, modelos Seq2Seq com redes recorrentes falham em capturar longas dependências textuais, devido ao gargalo de informação contido na representação a partir de um único vetor de contexto.

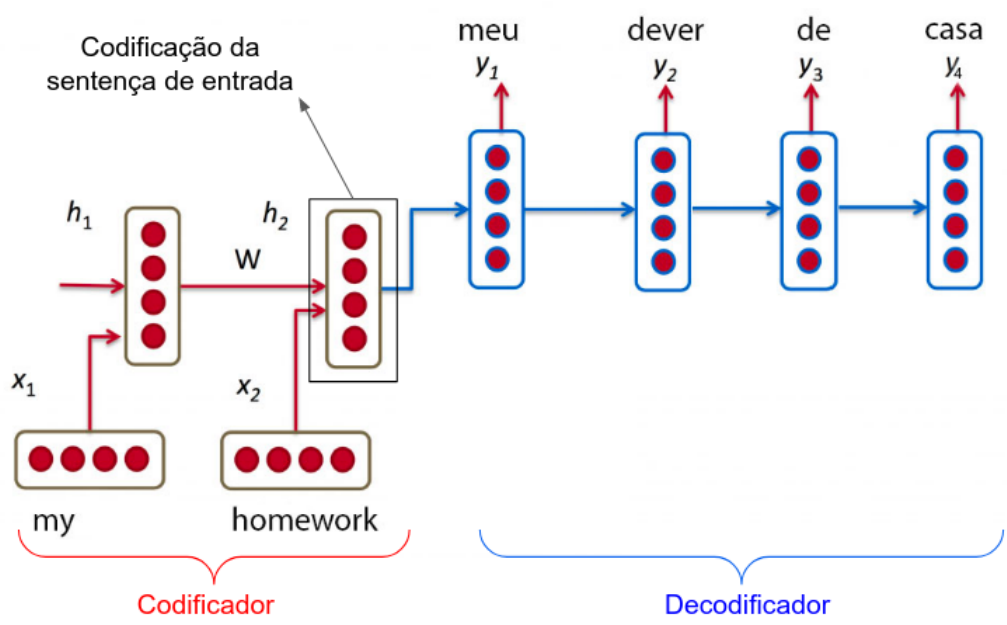

Figura 2.1: Ilustração de uma rede recorrente Seq2Seq na tarefa de Tradução Automática - Adaptado de [16]

2.3 Mecanismos de Atenção e Arquitetura Transformers

A motivação para a proposta apresentada em [2] foi perceber a deficiência de redes recorrentes ao usar o estado oculto final do codificador como o único vetor de contexto a ser usado pelo decodificador, na geração de um texto resultante de uma tarefa de tradução. Realmente, é difícil assumir que um único vetor consiga condensar todas as informações mais importantes referentes a uma sequência de entrada, principalmente quando a sentença de entrada é longa e composta de vários tokens.

Quando escutamos, por exemplo, "O CARRO ESTÁ NA OFICINA.", não damos a mesma importância a todas as palavras. Certamente, as palavras "carro" e "oficina" são as mais importantes para o receptor da mensagem. Mais do que isso, diferentes trechos da saída podem considerar diferentes trechos da entrada como relevantes.

Nessa linha, uma nova arquitetura de rede neural surgiu para abordar o problema de gargalo de informação presente nas redes neurais recorrentes. Trata-se das redes neurais com mecanismos de atenção. Tais mecanismos abordam o gargalo de informação de se usar apenas um vetor de contexto ao final do processo de codificação. Ao invés, o vetor de atenção fornece à rede decodificadora a visão inteira da sequência de entrada, em todos os passos do processo de decodificação. Assim, no processo de geração da saída, o

decodificador pode decidir quais palavras (tokens) são importantes a qualquer momento. A Figura 2.2 mostra como redes de atenção conseguem melhorar o desempenho de modelos neurais de tradução de texto, principalmente quando a sequência é longa.

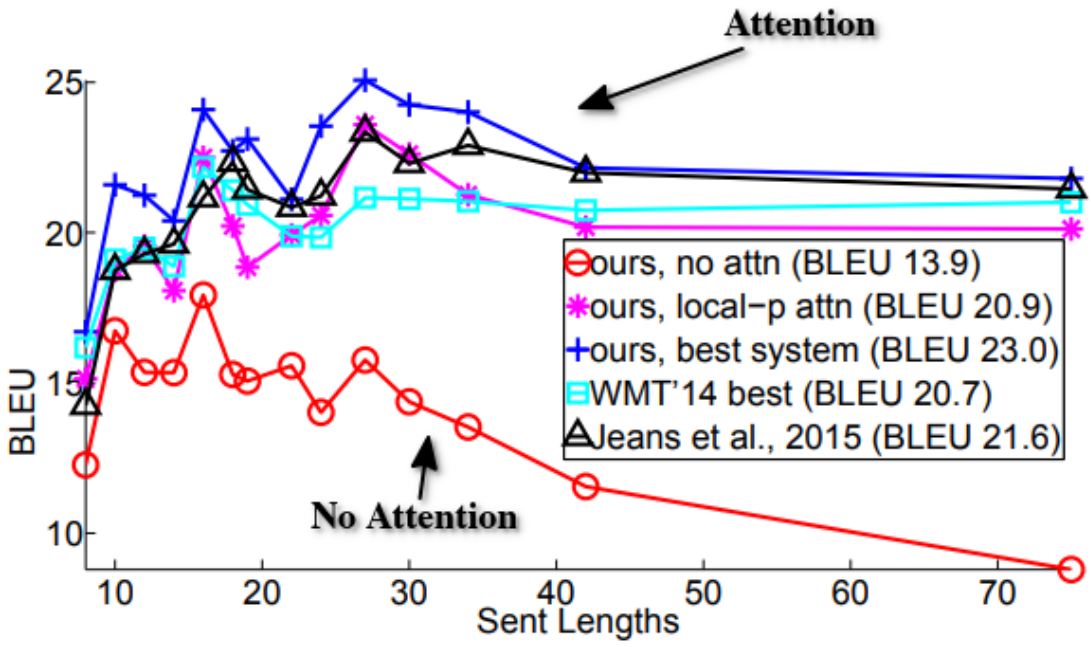

Figura 2.2: Desempenho em sentenças longas de alguns modelos na tarefa de tradução - Imagem de Luong et al. [62]

As redes com mecanismos de atenção trouxeram grandes avanços para os modelos de aprendizado de máquina, em especial para a área de PLN. Elencam-se alguns:

- Alternativa ao problema de gargalo de informação: mecanismos de atenção permitem que o decodificador olhe diretamente para a entrada, ultrapassando o gargalo.
- Ajuda no problema da instabilidade dos gradientes, especialmente se a sentença for longa: através de ligações diretas entre o decodificador e o codificador, fornecem atalhos entre estados distantes.
- Fornecem alguma interpretabilidade: inspecionando a distribuição de probabilidade da atenção, pode-se verificar em quais tokens da entrada o decodificador está focando.

Além disso, atenção é uma técnica geral de aprendizado profundo, e pode ser usada em várias arquiteturas (não só em modelos Seq2Seq) e em várias tarefas.

Fazendo uso somente de mecanismos de atenção, e dispensando componentes recorrentes e convolucionais, Vaswani *et al.*[95] criou uma arquitetura nomeada de *Transformer*, que obteve sucesso em inúmeras tarefas de PLN [17, 80]. Como os demais modelos Seq2Seq, a arquitetura *Transformer* também é composta de um codificador e de um decodificador. O codificador é composto de uma pilha com seis camadas idênticas. Cada camada possui duas sub-camadas. A primeira é uma rede de auto-atenção com múltiplas cabeças (*heads*). A segunda é uma rede neural simples que leva em consideração a posição de cada token na sequência. Conexões residuais foram empregadas nas duas subcamadas, seguidas de uma camada de normalização. Dessa forma, a saída de cada subcamada é $LayerNorm(x + Sublayer(x))$, onde $Sublayer(x)$ é a função de fato implementada pela subcamada. O decodificador também é composto de seis camadas idênticas. Em adição às duas subcamadas existentes no codificador, inseriu-se uma terceira camada no decodificador, a qual realiza atenção com múltiplas cabeças sobre a saída da última camada do codificador. Como no codificador, usam-se conexões residuais seguidas de uma camada de normalização. A camada de atenção é modificada com mascaramento para prevenir que posições intermediárias prestem atenção em posições futuras da sequência. A figura 2.3, extraída de [95], ilustra essa arquitetura.

2.4 Modelos de Linguagem Autorregressivos e Mascarados

De uma maneira ampla, modelos de linguagem computam a probabilidade de ocorrência de uma palavra dado um contexto de palavras. O modelo de linguagem tradicional, também chamado de autorregressivo, nos fornece a probabilidade de ocorrência de uma palavra, dadas as palavras anteriores. Dessa forma, dada uma sequência $\mathbf{x} = (x_1, x_2, \ldots, x_m)$, um modelo de linguagem autorregressivo tem como objetivo fornecer a probabilidade $p(x_t|x_{<t})$. A probabilidade de uma sequência de m palavras $x_1, x_2, \ldots, x_m$ é dada por $P(x_1, x_2, \ldots, x_m)$. Como o número de palavras que vem antes de uma palavra x_i varia de acordo com a sua localização na frase de entrada, $P(x_1, x_2, \ldots, x_m)$ é, às vezes, condicionada a uma janela de n palavras anteriores, ao invés de todas as anteriores. Nesses casos, tem-se que:

$$P(x_1, x_2, \ldots, x_m) = \prod_{i=1}^{m} P(x_i|x_1, \ldots, x_{i-1}) \approx \prod_{i=1}^{m} P(x_i|x_{i-n}, \ldots, x_{i-1}) \tag{2.1}$$

Há diferentes formas de produzir modelos de linguagem. Os modelos de linguagem

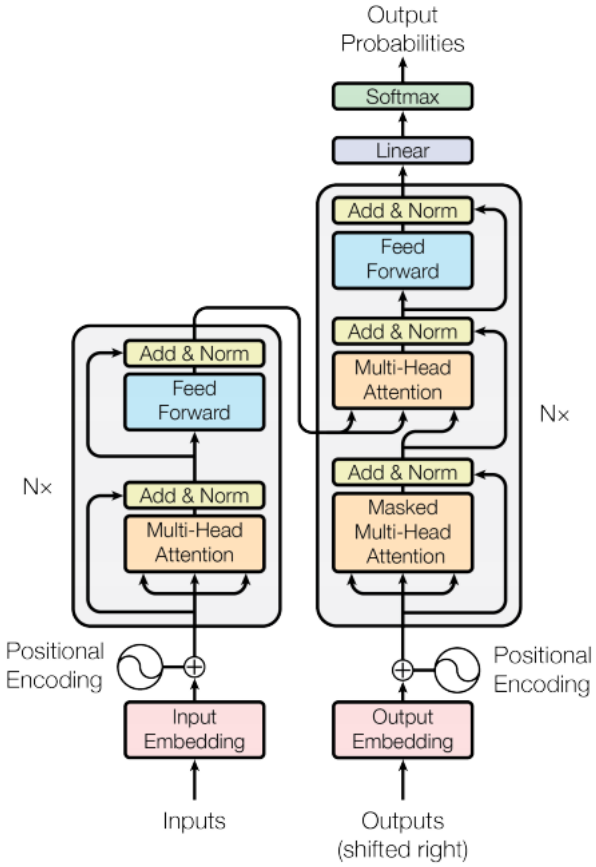

Figura 2.3: Arquitetura Transformer extraída de [95]

probabilísticos n-gram, por exemplo, usam contagem para calcular a probabilidade. Se um modelo usa tri-grams, a probabilidade de uma palavra dadas as duas anteriores é obtida pelo cálculo da frequência de cada tri-gram, dividida pela frequência dos bi-grams correspondentes:

$$P(x_3|x_1, x_2) = \frac{count(x_1, x_2, x_3)}{count(x_1, x_2)} \tag{2.2}$$

Modelos de linguagem n-gram possuem alguns pontos fracos. O primeiro é a esparsidade provocada por eventuais zeros no numerador e no denominador, uma vez que é necessário combinar todas as possibilidades de palavras do vocabulário. O outro problema surge da necessidade de armazenar todos os n-grams. Com isso, aumentando o n, o tamanho do modelo também aumenta.

Os modelos de linguagem que usam redes neurais são os que mais interessam ao presente estudo, e são os que têm alcançado o estado da arte em diversas tarefas de PLN. O treinamento de um modelo de linguagem neural envolve o aprendizado autosupervisionado de representações, onde, ao fim do treinamento, o modelo será capaz de

gerar representações contextualizadas dos tokens de uma sequência.

O uso de modelos de linguagem pré-treinados a partir de um grande volume de textos tem se mostrado extremamente efetivo para melhorar o desempenho de várias tarefas de PLN[75, 80, 17, 39]. O pré-treinamento consiste em treinar um modelo de linguagem em uma quantidade imensa de textos para obter representações contextualizadas. As tarefas incluem tarefas no nível de sentenças, como a de inferência de linguagem natural [101] e a de parafrasear [18], as quais buscam prever o relacionamento entre sentenças, assim como tarefas no nível de palavras (tokens), como a de responder perguntas automaticamente, em que modelos são treinados para produzir saídas refinadas no nível de palavras (tokens) [82].

Há várias formas de se aplicar as representações obtidas por modelos de linguagem pré-treinados a outras tarefas. Uma delas, como a utilizada pelo ELMo [75], usa arquiteturas específicas para cada tarefa e utiliza as representações pré-treinadas obtidas através do modelo de linguagem como atributos adicionais da tarefa. Outras abordagens, tais como fazem o OpenAI GPT [80] e o BERT [17], introduzem uma quantidade mínima de novos parâmetros para a tarefa específica e o treinamento consiste em fazer um ajuste fino de todos os parâmetros que já foram pré-treinados.

Durante o pré-treinamento de um modelo de linguagem, diferentes objetivos auto-supervisionados foram explorados na literatura. Dentre eles, o modelo de linguagem autorregressivo e o modelo de linguagem mascarado foram os que obtiveram mais sucesso [109, 110, 17, 54]. Os modelos de linguagem autorregressivos se baseiam nos mais tradicionais modelos de linguagem e buscam estimar a distribuição de probabilidade de um texto de uma maneira autorregressiva. Especificamente, dada uma sequência $\mathbf{x} = (x_1, x_2, \ldots, x_m)$, um modelo de linguagem autorregressivo fatora a probabilidade em um produtório da esquerda para direita $p(\mathbf{x}) = \prod_{t=1}^{T} p(x_t|x_{<t})$ ou da direita para esquerda $p(\mathbf{x}) = \prod_{t=T}^{1} p(x_t|x_{>t})$. O grande problema desses modelos é a unidirecionalidade, uma vez que, durante o treinamento, um token só pode considerar os tokens a sua esquerda ou a sua direita como contexto, dependendo do caso. Muitos trabalhos apontam que é fundamental para diversas tarefas obter representações que incorporem o contexto tanto da esquerda quanto da direita [17], ou seja, representações bidirecionais.

É razoável pensar que um modelo profundo bidirecional tem o potencial de alcançar melhor desempenho que um modelo unidirecional da esquerda para direita, ou até mesmo que a concatenação de um modelo da esquerda para direita com um da direita pra esquerda. Modelos autorregressivos tradicionais só podem ser treinados da esquerda para

direita ou da direita para esquerda, pois, se condicionássemos a distribuição a todos os tokens da sequência, permitiríamos que cada token se visse, tornando trivial a previsão da palavra alvo.

Assim, com o intuito de obter representações contextualizadas mais ricas em informação, Devlin *et al.*[17] introduziu a tarefa de modelagem de linguagem mascarada, que consiste, basicamente, em substituir uma porcentagem (no trabalho original, 15%) dos tokens da sequência por um token [MASK] e depois prever esses tokens mascarados. A arquitetura definida foi nomeada de BERT [17] e consiste basicamente do codificador de um *Transformer* (componente esquerdo da Figura 2.3). Assim, como saída principal desses modelos, obtém-se uma distribuição de probabilidade desse tokens mascarados, a qual leva em conta tanto o contexto esquerdo da frase, quanto o direito. O pré-treinamento dessas redes neurais profundas levou a uma série de avanços no aprendizado de representações de língua. Muitas tarefas não-triviais de PLN, incluindo aquelas com dados limitados, se beneficiaram desses modelos pré-treinados. Um sinal dessa evolução, por exemplo, é o desempenho desses modelos em uma tarefa de compreensão textual desenvolvida para exames de inglês do nível fundamental e médio na China, o teste RACE [52]. Enquanto o primeiro modelo que descreveu a tarefa e formulou o desafio obteve uma acurácia de $44,1\%$, o modelo de linguagem mascarado (MLM) ALBERT [54] obteve $89,4\%$. Pontua-se que, em modelos mascarados como BERT e ALBERT, somente os tokens mascarados são reconstruídos durante o treinamento, e não a entrada inteira, como no caso de modelos autorregressivos.

2.4.1 ALBERT - Um BERT mais leve

De um modo geral, o aumento do tamanho de um modelo de linguagem neural pré-treinado tem um impacto positivo em tarefas subsequentes. Em algum ponto, entretanto, fica inviável o crescimento do modelo, devido a limitações de memória do GPU/TPU e também a enormes tempo de treinamento. Nesse sentido, ALBERT [54] endereçou esses problemas com duas técnicas de redução de parâmetros. Nessa dissertação, usa-se um modelo ALBERT com o objetivo de se extrair conhecimento das ricas representações contextualizadas que ele é capaz de gerar, devido a seu pré-treinamento em uma enorme quantidade de dados. Apesar de a adoção de um outro MLM ser possível, optou-se pelo ALBERT pelo fato de usar menos recursos computacionais, quando comparado com um BERT com o mesmo tamanho.

O esqueleto arquitetural do ALBERT é similar ao do BERT, o que significa que ele

usa um codificador *Transformer* [95] com função de ativação GELU [32]. Seguindo a mesma notação do BERT, indicam-se o tamanho das representações vetoriais dos tokens do vocabulário como E, o número de camadas do codificador como L e o tamanho das representações das camadas ocultas como H. Há três contribuições principais que ALBERT faz sobre as escolhas de projeto do BERT.

Fatoração dos Parâmetros. No BERT, o tamanho das representações dos tokens do vocabulário E está amarrado ao tamanho das representações dos estados ocultos H, ou seja, $E = H$. Em [54] argumenta-se que essa vinculação não é eficiente, tanto por razões de modelagem quanto por razões práticas. Do ponto de vista da modelagem, as representações dos tokens do vocabulário que são aprendidas são representações que independem do contexto, enquanto que as representações das camadas ocultas aprendidas são representações que dependem do contexto. Como o poder representacional do BERT reside na possibilidade de se obter representações ricas contextualizadas a partir de representações não-contextualizadas, desamarrar o tamanho das representações dos tokens do vocabulário E do tamanho das representações dos estados ocultos H torna mais eficiente o uso dos parâmetros do modelo. Do ponto de vista prático, como em tarefas de processamento de língua natural o tamanho do vocabulário V costuma ser grande, se $E = H$, aumentando-se o tamanho H naturalmente aumenta-se a matriz de representações de tokens do vocabulário, a qual tem tamanho $V \times E$. Isso pode resultar em um modelo com bilhões de parâmatros, muitos dos quais são atualizados esparsamente durante treinamento. Assim, no ALBERT, a matriz das representações do vocabulário é decomposta em duas matrizes menores, reduzindo os parâmetros de $O(V \times H)$ para $O(V \times E + E \times H)$. Essa redução é significativa quando $H \gg E$.

Compartilhamento de Parâmetros entre Camadas. ALBERT foi proposto com uma esquema de compartilhamento de parâmetros entre as camadas, com o intuito de melhorar a eficiência dos parâmetros. Apesar de ser possível compartilhar parcialmente os parâmetros entre as camadas, a configuração padrão adotou o compartilhamento de todos os parâmetros, tanto os das redes neurais quantos os parâmetros referentes às redes de atenção. Com isso, é possível aumentar a profundidade, sem aumentar o número de parâmetros.

Componente de Coerência entre Sentenças na Função de Custo. Assim como o BERT, ALBERT também usa um componente extra na função de custo, além

do componente cujo objetivo é prever os tokens mascarados. No BERT, esse componente é chamado de *next-sentence prediction*. Durante o treinamento, esse componente busca aprender se dois trechos de sentença aparecem consecutivamente no texto original, e tem como objetivo melhorar o desempenho do modelos em tarefas subsequentes que requeiram raciocínio sobre o relacionamento de dois pares de sentença. Argumentando que esse componente não é efetivo devido à sua falta de dificuldade como tarefa, quando comparado com a tarefa do modelo de linguagem mascarado, Lan *et al.* [54] propõe um componente chamado de *sentence-order prediction* (SOP). Esse componente usa como exemplo positivo dois segmentos de textos consecutivos do mesmo documento (como BERT) e como exemplo negativo os mesmos dois segmentos, mas com a ordem trocada. Os resultados indicaram que ALBERT melhorou o desempenho em tarefas subsequentes que envolvem codificação de mais de uma sentença.

Apesar de o uso de modelos pré-treinados mascarados, como BERT e ALBERT, ter obtido grande sucesso em tarefas ditas como de entendimento da língua, a aplicação deles diretamente em tarefas de geração de textos, não é factível, uma vez que eles são modelados para tarefas que usam somente um codificador, tornando-os incompatíveis com modelos de geração de textos, que requerem um decodificador. Entretanto, é comum ao escrever um texto que palavras que surgiram antes possam ser alteradas depois de se escrever uma sequência posterior, dando uma ideia intuitiva de contexto e bidirecionalidade. Com intuito de fazer uso de ricas representações bidirecionais provenientes de um modelo de linguagem mascarado, esta dissertação busca destilar conhecimento dele.

2.4.2 Destilação de Conhecimento

Destilar consiste em extrair o conhecimento contido em outro modelo por meio de uma técnica específica de treinamento. Essa técnica é normalmente usada para transferir informações contidas em um modelo grande já treinado, chamado de professor, para um modelo menor, chamado de estudante, o qual é mais adequado para ser colocado em produção. Diferente de métodos como [33, 91], em que tanto o professor como o estudante são treinados na mesma tarefa e o objetivo principal da destilação é comprimir conhecimento em um modelo menor, a proposta dessa dissertação é usar a destilação para se aproveitar das representações pré-treinadas bidirecionais geradas por meio de um modelo de linguagem mascarado. Assim, ALBERT fornece rótulos mais suaves para serem usados como alvos durante o treinamento, no componente de destilação de conhecimento da função de custo do modelo Seq2Seq, tendo como inspiração o método proposto em

[12], com intuito de melhorar a qualidade do texto gerado.

2.5 Trabalhos Relacionados

A tarefa de Transferência de Estilo de Texto (TST) tem como objetivo alterar de forma automática traços estilísticos de um texto, como formalidade, sentimento, estilo autoral, humor, complexidade, entre outros, mas tentando garantir a preservação de seu conteúdo. Em [40] é introduzida uma taxonomia para a tarefa de TST de acordo com a abordagem adotada para resolver a tarefa de transferência de estilo. Para posicionar as contribuições desta dissertação e apresentar a literatura existente sobre a tarefa de TST não-supervisionada, vamos a adotar a segmentação para esse tipo de tarefa adotada por [40]. Apesar de o citado estudo se preocupar com a tarefa de transferência de estilo de forma ampla, tratando também de configurações onde há dados paralelos e até aquelas onde sequer é sabido o domínio da sentença textual, essa dissertação só se baseou em como a tarefa de TST não-supervisionada foi dividida pelo estudo. Pontua-se que [40] chamou de **transferência de estilo textual não-paralelo supervisionada** a tarefa que essa dissertação e a maioria dos trabalhos relacionados chamou de **transferência de estilo textual não-supervisionada**. Entende-se não ser importante essa discussão sobre a nomenclatura adotada e reforça-se que essa dissertação endereça o problema de transferência de estilo textual não-supervisionada, conforme já definido anteriormente. A Figura 2.4 mostra como os modelos que executam esse tipo de tarefa podem ser divididos em três macro-grupos, conforme proposto por [40].

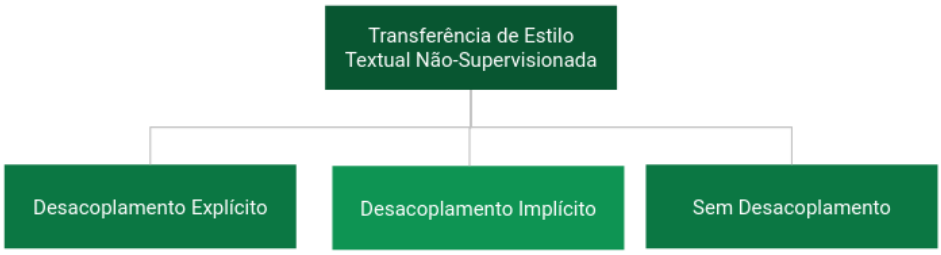

Figura 2.4: Taxonomia da tarefa de TST Não-Supervisionada

Esta dissertação busca criar um modelo treinado com um método de aprendizado de máquina que tenha bom desempenho na tarefa de TST, supondo a ausência de dados paralelos durante o treinamento, ou seja, o foco é em aprendizado de máquina não-supervisionado. Dessa forma, apesar de a dissertação não ter como foco principal uma revisão da literatura para a tarefa de TST não-supervisionada, as principais estratégias

adotadas pela literatura serão comentadas a seguir, para posicionar a contribuição da dissertação frente à literatura existente. Foram conduzidas buscas nas principais bases digitais acadêmicas,a saber, nas bases *Scopus*, *Science Direct* e *IEEE*, e também nas principais conferências relacionadas à área de PLN (ACL, EMNLP, HLT-NAACL, COLING e EACL), sempre buscando trabalhos no tema de Transferência de Estilo Textual (TST) Não-Supervisionada publicados nos últimos cinco (5) anos.

Serão analisados os três tipos de estratégias para desacoplar estilo de conteúdo, a saber o desacoplamento implícito, o desacoplamento explícito e as abordagens que não tentam desacoplá-los, conforme [40]. Como veremos, a maioria dos estudos buscam, a partir de um codificador, obter representações do conteúdo da sentença desacopladas do estilo. Após isso, o decodificador recebe como entrada essa representação junto com uma representação do estilo alvo, para gerar uma variação da sequência de entrada com o estilo desejado. Tal desacoplamento não é fácil de ser obtido e [53] mostra que tampouco é necessário para a tarefa TST. A abordagem adotada nessa dissertação, que será pormenorizada no próximo capítulo, não faz uso de desacoplamento.

Na sub-seção 2.5.1, discorre-se sobre esses trabalhos relacionados e mostra-se como esses trabalhos estão distribuídos de acordo com as tarefas executadas e conforme a presença ou não de algumas características no modelo, comparando-os ao método proposto nessa dissertação.

2.5.1 Estudos conforme a Abordagem de Desacoplamento

Os trabalhos relacionados à tarefa de Transferência de Estilo Textual (TST) que não usam dados paralelos podem ser divididos em três (3) macro-grupos, de acordo com a abordagem teórica adotada para desacoplar conteúdo e estilo da sentença original. O primeiro grupo são os trabalhos que executam a tarefa editando explicitamente determinados tokens da sentença que indicam o estilo dela. Essa abordagem gera textos através da **substituição explícita** de trechos da sequência de entrada. Fazem parte do segundo grupo os modelos que buscam separar conteúdo e estilo da sentença original, mas não de forma explícita alterando a sentença original. Para estes modelos, é dito que há um **desacoplamento implícito** entre conteúdo e estilo da sentença, o qual acontece de forma puramente teórica. Nessa abordagem, para separar conteúdo do estilo, os modelos aprendem representações do conteúdo e do estilo para um dado texto. Em seguida, a representação do conteúdo é combinada com a representação do estilo alvo para gerar o texto no estilo alvo. O terceiro e último grupo de modelos são aqueles que **não procuram desacoplar** o estilo do

conteúdo, nem explicitamente através da edição do texto de entrada, nem teoricamente por meio da criação de variáveis ocultas latentes que representem somente o estilo e somente o conteúdo. Estudo recente mostra que é difícil julgar se representações de conteúdo e estilo obtidas estão de fato desacopladas [53]. Nessa linha, estudos mais recentes, incluído esta dissertação, exploram a tarefa de TST sem procurar desacoplar conteúdo e estilo.

2.5.1.1 Desacoplamento Explícito

Nessa estratégia, separa-se explicitamente conteúdo de estilo, através da substituição de palavras-chave que são associadas a determinado estilo. Nessa linha, elencam-se [56, 90, 104, 106, 115, 105, 63].

Em [56] foi proposto o método *Delete, Retrieve, Generate* que faz explicitamente a substituição de palavras-chave de um texto por palavras do estilo alvo. O método funciona da seguinte forma: primeiramente, por meio de estatísticas, encontram-se e deletam-se as palavras que mais representam o estilo original. O texto original subtraído das palavras deletadas é chamado de conteúdo. No próximo passo, o texto que é mais similar à entrada é buscado no *corpus* de destino. As palavras mais associadas ao estilo de destino são extraídas do texto retornado e combinadas com o conteúdo para gerar o texto de saída, usando um modelo de redes neurais *Sequence-to-Sequence*. Em [115], adotou-se uma técnica similar de substituição de palavras-chave para realizar transferência de sentimento em textos. Ainda, em [90] o modelo *Delete, Retrieve, Generate* foi estendido para melhorar a etapa de *Delete* usando Transformer [95].

Em [104] foi proposto o método *POINT-THEN-OPERATE* que também faz operações na sentença de entrada e é baseado em aprendizado hierárquico por reforço. De uma maneira iterativa, um agente de alto nível indica as posições a serem editadas da sentença, e um agente de baixo nível altera a sentença baseado nas indicações de alto nível. Na tarefa de transferência de sentimento, [106] propôs um modelo com um módulo de *neutralização* e outro de *emocionalização*. O módulo de neutralização é responsável por extrair informação semântica não emocional por meio da filtragem explícita de palavras (tokens) emocionais. O módulo de emocionalização é responsável por adicionar sentimento ao conteúdo semântico neutralizado proveniente do módulo inicial.

Em [105] foi proposto um modelo em duas etapas chamado *Mask and Infill*: na primeira etapa, de mascaramento, palavras associadas ao estilo são mascaradas usando um método de taxa de frequência. No passo de preenchimento, um modelo de linguagem pré-treinado é usado para preencher as posições mascaradas, prevendo palavras ou frases

no estilo de destino. O método apresentado em [63] também usa um modelo de linguagem mascarado pré-treinado para encontrar os trechos a serem removidos e também para gerar os trechos substitutos. É importante pontuar que a maneira que [105, 63] usam o modelo de linguagem mascarado é completamente diferente da maneira que usamos no método proposto nesta dissertação. Enquanto esses outros trabalhos usam um modelo de linguagem mascarado para prever os tokens que foram previamente selecionados para substituição, em nosso modelo o principal papel do modelo de linguagem mascarado é suavizar as probabilidades usadas para gerar o texto, almejando a geração de textos mais semelhantes ao estilo desejado.

2.5.1.2 Desacoplamento Implícito

Diversas técnicas, como *back-translation*, aprendizado adversarial e geração controlada, são usadas para desacoplar e extrair representações do conteúdo e do estilo de uma sequência.

Para obter representações do conteúdo de uma sentença agnósticas ao estilo, a maioria das soluções usa aprendizado adversarial [22, 41, 88, 44, 118, 10, 51, 60, 112]. Após o aprendizado da representação oculta latente ao conteúdo, o decodificador, que é o componente responsável por gerar a sentença, recebe como entrada a representação oculta junto com o rótulo do estilo desejado para gerar uma variação do texto de entrada com o estilo desejado. Em [110] dois modelos de linguagem, um para cada domínio estilístico, são usados durante o treinamento, e o modelo minimiza a perplexidade das sentenças geradas de acordo com esses modelos de linguagem pré-treinados.

Outra técnica usada para desacoplar conteúdo de estilo é aprender representações da entrada em um outro domínio e depois convertê-las de volta para o domínio original, forçando-as a serem iguais à entrada. Essa técnica é inspirada no arcabouço *back-translation*, vastamente usado para tarefas de tradução e criação de dados artificiais (*data augmentation*). Nessa linha, em [78] um modelo neural de tradução de textos inglês-francês é aprendido para refrasear a sentença e remover as propriedades estilísticas do texto. Nesse modelo, inicialmente a frase em inglês é traduzida para o francês, usando o modelo neural de tradução. Depois, o texto em francês é traduzido de volta para o inglês usando um modelo neural francês-inglês. A representação latente aprendida usando o modelo tradutor, teoricamente, contém somente informações de conteúdo. Por fim, a representação latente aprendida é usada para gerar textos em um estilo diferente, usando uma abordagem com múltiplos decodificadores. O trabalho apresentado em [116] também

faz uso de *back-translation* em sua modelagem. Usando um arcabouço estatístico para tradução, inicialmente são criados dados pseudo-paralelos. Após essa fase, esses dados são usados para inicializar um *pipeline* iterativo de *back-translation* e treinar dois sistemas de transferência de estilo baseados em modelos neurais de tradução.

Trabalhos existentes que endereçaram a tarefa de TST também exploraram a estratégia de aprender um atributo de estilo para controlar a geração de textos em diferentes estilos. Diferente de *autoencoders*[35], que aprendem representações comprimidas do dado de entrada, abordagens baseadas em *Variational Autoencoder* (VAE) [49] aprendem parâmetros de uma distribuição de probabilidade que representam os dados. A distribuição também pode ser usada para gerar amostras da distribuição. Dessa forma, a natureza generativa dos VAEs os torna bastante usados em diversas tarefas de geração de textos em língua natural [23]. O método apresentado em [41] induz um modelo que faz uso de VAE para aprender representações latentes das sentenças. Essas representações são compostas de variáveis não-estruturadas (conteúdo) z e de variáveis estruturadas (estilo) c que têm como objetivo representar características salientes e independentes da semântica da sentença. Por fim, z e c são inseridos em um decodificador para gerar textos no estilo desejado. O método apresentado em [94] estendeu essa abordagem, adicionando restrições para preservar um conteúdo independente do estilo, usando informação de categorias gramaticais (*Part-of-speech*) e um modelo de linguagem condicionado ao conteúdo. Já em [117], foi proposto um *autoencoder* adversarial regularizado, expandindo o uso de *autoencoder* adversariais para sequências discretas. Por fim, em [73], fazendo uso de treinamento adversarial e de VAE, métodos anteriores de geração de paráfrases diversas foram expandidos para permitir que essa geração seja guiada por um estilo alvo.

2.5.1.3 Abordagens Sem Desacoplamento

Apesar de as técnicas inseridas nessa abordagem não pressuporem a necessidade de desacoplar conteúdo e estilo das sentenças de entrada, o mesmo conjunto de técnicas de aprendizado de máquina, como geração controlada, aprendizado adversarial, aprendizado por reforço e modelos probabilísticos também são usadas.

Em [42] foi criado um arcabouço para transformação controlada de linguagem natural. O núcleo do arcabouço consiste de uma rede neural do tipo codificador-decodificador, a qual recebe o reforço de conhecimento das transformações que vão sendo realizadas através de módulos auxiliares. Em [114] foi apresentado o método SHAPED (*shared-private encoder-decoder*). A arquitetura do SHAPED possui parâmetros compartilhados

que são atualizados com base em todos os exemplos de treinamento, assim como possui parâmetros privados que são atualizados somente com exemplos de suas respectivas distribuições. Em [119] foi proposto um método sequência-para-sequência (Seq2seq) com atenção que dinamicamente avalia a relevância de cada palavra de saída para o estilo alvo. O método proposto em [53] também usou a estratégia de aprender representações dos atributos para controlar a geração de texto, no entanto sem tentar desacoplar conteúdo e estilo. Nesse trabalho, também foi demonstrado que é difícil provar que o estilo está realmente fora da representação de conteúdo desacoplada obtida e que também não é necessário realizar esse desacoplamento para que a tarefa de TST seja bem sucedida. Em [15] foi proposto um método que aprende um modelo baseado em Transformer [95] com os vetores de estilo sendo parâmetros treináveis do modelo. O trabalho proposto nesta dissertação se vale dessa arquitetura para a tarefa de transferência de etilo. Entretanto, o método proposto nesta dissertação difere na estratégia de treinamento, uma vez que altera a função de custo da rede geradora, para extrair conhecimento de um modelo de linguagem mascarado.

O método proposto em [69] é composto por um VAE (*Variational Autoencoder*) recorrente e um módulo de rede neural preditora da saída. Ao impor condições de contorno durante a otimização e usar o decodificador do VAE para gerar as sentenças revisadas, o método garante que a transformação é similar à sentença original, está associada a melhores saídas e parece natural. Já o método proposto [57] possui três componentes: 1) um VAE [49], que possui um codificador que mapeia a sentença para um espaço vetorial contínuo e suave e um decodificador que mapeia de volta a representação contínua para uma sentença; 2) preditores de atributos, que usam a representação contínua obtida pelo VAE como entrada e preveem os atributos da sentença de saída; e 3) preditores de conteúdo, que buscam prever uma variável *Bag-of-word* (BoW) para a sentença de saída. Em [96], foi apresentado um método que primeiro compreende um *autoencoder* baseado em Transformers, que tem como objetivo aprender uma representação oculta da entrada e, após esse aprendizado, transforma a tarefa em um problema de otimização que edita a representação oculta obtida, até ela se adequar ao atributo alvo. Em [107] também é usado um VAE e condições de contorno são impostas à distribuição de probabilidade obtida, para tornar possível uma geração de texto controlada. Os trabalhos apresentados em [69, 57, 96, 107] têm em comum o fato de manipularem as representações ocultas obtidas da sentença de entrada para geração de textos no estilo desejado.

O método proposto em [24] aborda a tarefa de TST usando uma arquitetura de aprendizado por reforço composta por uma rede geradora e outra avaliadora. A sentença é

gerada usando um codificador-decodificador com atenção. O avaliador é um discriminador de estilo treinado adversarialmente com restrições semânticas e sintáticas que pontuam a sentença gerada pelo estilo, preservação de conteúdo e fluência. O método proposto em [61] também usa aprendizado por reforço e considera o problema de transferir o estilo de um domínio para o outro e vice-versa como uma tarefa dual. Para tanto, duas recompensas são modeladas baseadas nessa estrutura, para refletir o controle do estilo e a preservação do conteúdo.

O método proposto em [31] aborda a tarefa de TST com aprendizado não-supervisionado, formulando-a como um modelo generativo profundo probabilístico, onde o objetivo de otimização surge naturalmente, sem a necessidade de criação de objetivos customizados artificiais.

Há diversas tarefas que podem ser abordadas usando modelos de transferência de estilo. Citam-se, como aplicações mais modernas, as tarefas de tradução de linguagem ofensiva em mídias sociais [19], de construção de um parágrafo coerente a partir de *tweets* desconexos de determinado domínio [1], de geração de sarcasmo [67] e de correção de textos não-fluentes [86].

Nessa linha, em [50] a tarefa de transferência de estilo (não-supervisionada) foi reformulada como um problema de geração de paráfrases. Por meio do ajuste fino de modelos de linguagem pré-treinados em paráfrases geradas automaticamente, o método proposto alcançou ótimos resultados na métrica de similaridade semântica [99], nas tarefas de imitação autoral e de fusão de sentenças. O método proposto em [58] também usa a mesma métrica de similaridade para avaliar a geração de textos, porém o modelo induzido otimiza, durante o treinamento, funções de recompensa que explicitamente consideram diferentes aspectos estilísticos da sentença transferida.

O método proposto em [46] usa um *autoencoder* baseado em Transformers para reconstruir a sentença original, e uma representação adaptativa para o estilo, a qual é aprendida em um módulo próprio de estilo. Ao separar em dois módulos, o intuito é que cada um foque melhor em sua tarefa específica. Ao invés de criar representações vetoriais para representar os estilos a partir de uma única sentença, o método proposto em [111] usa um técnica chamada *generative flow* para extrair propriedades estilísticas a partir de múltiplas instâncias de cada estilo, as quais formam um espaço estilístico oculto mais expressivo e discriminativo.

Em [11] foi apresentado um método baseado em um Tranformer Não-Autorregressivo (NAT) [29]. O método consiste em duas partes: um módulo que independe do estilo, com

parâmetros compartilhados entre todos os estilos, e um módulo dependente do estilo. A abordagem proposta em [27] inicializa um codificador-decodificador com um modelo de linguagem baseado em Transformer pré-treinado em um corpus genérico, e aumenta sua capacidade de reescrever em múltiplos estilos, ao usar modelos de linguagem atentos a cada estilo como discriminadores.

A Tabela 2.1 resume os trabalhos relacionados conforme a abordagem usada para desacoplar conteúdo do estilo na tarefa de TST Não-Supervisionada.

Tabela 2.1: Publicações baseadas na abordagem para desacoplar conteúdo e estilo

Abordagem	**Publicações**
Desacoplamento Explícito	[56, 90, 104, 106, 115, 105, 63]
Desacoplamento Implícito.	[22, 41, 88, 44, 118, 10, 51, 60, 112, 110, 78, 116, 41, 94, 117, 73]
Sem Desacoplamento	[42, 114, 119, 53, 15, 69, 57, 96, 107, 24, 61, 31, 50, 58, 46, 111, 11, 27], MATTES

Já a Tabela 2.2 apresenta a relação dos trabalhos existentes na literatura sobre o tema conforme as tarefas executadas e os componentes de treinamento presentes.

Neste capítulo foram discutidas as técnicas de aprendizado de máquina adotadas nesta dissertação, assim como os principais trabalhos recentes relacionados ao tema. No próximo capítulo, a abordagem para endereçar a tarefa de TST não-supervisionada usando um MLM para destilar conhecimento será pormenorizada.

Tabela 2.2: Publicações conforme as tarefas executadas

Abordagem	modelo	Tarefas						
		Imitação Autoral	Transferência de sentimento	Decifra-mento de palavras substituídas	Tradução de línguas parecidas	Tradução Não Supervisionada	Formalização	Outras
Desacoplamento Explícito	[56]		X					
	[90]		X					
	[104]		X					
	[106]		X					
	[115]		X					
	[105]		X					
	[63]		X					X
Desacoplamento Implícito	[22]		X					X
	[41]		X					X
	[88]		X	X				X
	[44]		X					
	[118]	X	X					X
	[10]		X	X				
	[51]		X					X
	[60]		X					X
	[112]		X					
	[110]		X	X	X			
	[78]		X					X
	[116]		X					
	[41]		X					X
	[94]		X					X
	[117]		X					X
	[73]							X
Sem Desacoplamento	[42]						X	
	[114]							X
	[119]		X				X	
	[53]		X					X
	[15]		X					
	[69]	X	X					
	[57]		X					X
	[96]		X					
	[107]		X					
	[24]		X				X	
	[61]		X				X	
	[31]	X	X	X	X	X		
	[50]	X					X	
	[58]		X				X	
	[46]		X					
	[111]		X				X	X
	[11]		X	X	X			
	[27]		X				X	
	MATTES	X	X					

Tabela 2.3: Publicações conforme as características do modelo

Abordagem	modelo	Características do Modelo						
		Controla Múltiplos Estilos	Usa Back-Translation	Treinamento Adversarial	Usa LM Pré-Treinado	Usa MLM Pré-treinado	Baseado em Transformers	Avaliação Humana
Desacoplamento Explícito	[56]							X
	[90]						X	X
	[104]				X			X
	[106]							X
	[115]							X
	[105]					X		X
	[63]					X		
Desacoplamento Implícito	[22]			X				X
	[41]							
	[88]			X				X
	[44]			X				X
	[118]			X				X
	[10]			X				X
	[51]	X		X				X
	[60]	X	X	X				X
	[112]			X	X			X
	[110]			X	X			
	[78]		X	X				X
	[116]		X					X
	[41]							
	[94]				X			X
	[117]			X				X
	[73]			X				
Sem Desacoplamento	[42]				X			X
	[114]							
	[119]				X			X
	[53]	X	X					X
	[15]		X	X			X	X
	[69]							
	[57]	X						X
	[96]						X	X
	[107]							
	[24]			X	X			X
	[61]							X
	[31]		X		X			
	[50]		X		X		X	X
	[58]			X	X		X	X
	[46]						X	
	[111]			X				X
	[11]	X		X			X	X
	[27]	X			X		X	X
	MATTES		X	X		X	X	X

Capítulo 3

MATTES: Uma Abordagem de Destilação de Conhecimento de Modelos Mascarados para Tranferência de Estilo

Neste capítulo, será apresentada a abordagem proposta para destilar conhecimento a partir de um modelo de linguagem mascarado, com o objetivo de melhorar a qualidade do texto gerado por um modelo Seq2Seq para a tarefa de transferência de estilo textual. O modelo de linguagem mascarado selecionado é o ALBERT [54], uma versão mais leve do popular BERT mas que ainda obtém resultados melhores ou competitivos, e que se vale de técnicas de redução de parâmetros para melhorar a eficiência do treinamento e a redução de custos de memória. O método proposto, chamado de **MATTES** (*MAsked Transformer for TExt Style transfer*), segue a arquitetura estabelecida em [15], que por sua vez se vale de um arcabouço de treinamento adversarial [79]. Assim, são usadas duas redes neurais durante o treinamento. Uma delas é uma rede discriminadora usada somente durante o treinamento. Essa rede é basicamente um classificador de estilos e é usada com o objetivo de que o modelo aprenda a diferenciar o estilo de sentenças presentes no conjunto de dados do estilo de sentenças geradas pelo modelo artificialmente. A outra rede neural é a geradora, responsável por gerar os textos de saída e composta de um codificador e um decodificador, baseada em uma arquitetura Transformer. A rede geradora recebe como entrada uma sentença X e um estilo de saída alvo s, e produz uma sentença Y no estilo alvo, tornando o modelo proposto uma função mapeadora $Y = f_\theta(X, s)$. A Figura 3.1 apresenta uma representação gráfica de como o MATTES opera. Salienta-se que o modelo final aprendido pelo MATTES e usado para inferência é a rede geradora, enquanto a rede discriminadora é usada somente durante o treinamento.

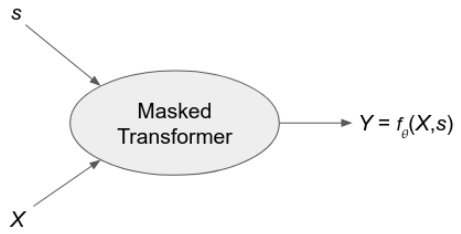

Figura 3.1: Transformer Mascarado

Antes de entrar em detalhes sobre os componentes específicos do MATTES, inicialmente, na Seção 3.1, será apresentada a formalização do problema investigado nesta dissertação, a saber, a tarefa de transferência de estilo textual (TST). Após, são apresentados os dois principais paradigmas em que o método se apoia: o aprendizado Seq2Seq na Seção 3.2, para que o modelo receba uma sentença de entrada e aprenda a devolver uma sentença de saída, e o modelo de linguagem mascarado na Seção 3.3, que é usado para obter uma distribuição de probabilidade para cada token da sentença. Após, na Seção 3.4, será apresentado o componente de destilação de conhecimento da função de custo, o qual é treinado usando uma configuração para extrair conhecimento do modelo de linguagem mascarado. Por fim, a Seção 3.5 descreve o algoritmo de aprendizado adversarial proposto aqui.

3.1 Formulação do Problema

Nesta dissertação, assume-se que os estilos são nomeados por elementos pertencentes a um conjunto S. Por exemplo, $S = \{positivo, negativo\}$ para a tarefa de transferência de sentimento, onde o estilo textual pode ser *positivo* ou *negativo*. Para o treinamento do modelo, assume-se acesso a um conjunto de sentenças com seus rótulos de estilo $D = \{(X_1, s_1), \ldots, (X_k, s_k)\}$, onde X_i é uma sentença e $s_i \in S$ é o atributo de estilo da sentença. A partir de D, define-se um conjunto de sentenças $D_s = \{X : (X, s) \in D\}$, as quais representam todas as sentenças pertencentes a D com atributo s. Por exemplo, na tarefa de transferência de sentimento, seriam todas as sentenças com atributo *positivo*. Para todas as sequências de um mesmo conjunto de dados D_i, assume-se que todas compartilham características específicas relacionadas ao estilo das sequências.

O objetivo principal da tarefa de aprendizado de transferência de estilo textual é construir um modelo que receba como entrada uma sentença X e um estilo alvo s^{tgt}, onde X é uma sentença com estilo $s^{src} \neq s^{tgt}$, e produza uma sentença Y que preserve o máximo possível o conteúdo de X enquanto incorporando o estilo s^{tgt}. O método MATTES pro-

posto nesta dissertação tem como objetivo abordar o problema de transferência de estilo com aprendizado de máquina não-supervisionado. Assim, pontua-se que os únicos dados disponíveis para treinamento são a sequência X e o seu respectivo estilo s^{src}. Não se tem acesso a uma sentença gabarito X^*, que seria a conversão de X para o estilo alvo s^{tgt}.

3.2 Arquitetura de Aprendizado Sequência-para-Sequência

Em aprendizado de máquina para PLN, o paradigma Seq2seq compreende o conjunto de modelos baseados em redes neurais que recebem como entrada uma sequência de palavras (ou seja, um texto ou uma frase) e geram outra sequência de palavras como saída. Uma rede neural composta de um componente codificador e de um componente decodificador é geralmente a arquitetura adotada para esse tipo de tarefa. Formalmente, durante o aprendizado, o modelo é treinado para gerar uma sequência de saída $Y = (y_1, \ldots, y_N)$ de comprimento N, condicionada à sequência de entrada $X = (x_1, \ldots, x_M)$ de comprimento M, onde $x_i \in X$ e $y_i \in Y$ são tokens. Essa rede neural alcança o objetivo de gerar a sequência de saída aprendendo durante o treinamento uma distribuição de probabilidade condicional $P_\theta(Y|X)$, ao minimizar a função de custo de entropia cruzada $\mathcal{L}(\theta)$, descrita na Equação 3.1, onde θ são os parâmetros do modelo:

$$\begin{aligned} \mathcal{L}(\theta) &= -log P_\theta(Y|X) \\ \mathcal{L}(\theta) &= -log \sum_{t=1}^{N} P_\theta(y_t|y_{1:t-1}, X) \end{aligned} \tag{3.1}$$

Há diversos modelos neurais que são treinados para gerar textos. Seguindo [15], nesta dissertação foi adotada a arquitetura *Transformer* [95] como o modelo Seq2Seq. Após o treinamento, durante o processo de inferência, ou seja, de conversão do estilo de uma sentença de entrada, além da sentença de entrada X, o modelo também recebe como entrada o estilo de saída desejado, que também é inserido no codificador junto com a sentença (Figura 3.1). Dessa forma, o método tem como objetivo aprender um modelo que representa uma distribuição de probabilidade, condicionada não só em X mas também no estilo alvo desejado s^{tgt}. Assim, as Equações 3.1 são modificadas para atenderem a essa característica, dando origem às Equações 3.2, conforme segue:

$$\begin{aligned} \mathcal{L}(\theta) &= -logP_\theta(Y|X,s) \\ \mathcal{L}(\theta) &= -log\sum_{t=1}^{N} P_\theta(y_t|y_{1:t-1},X,s) \end{aligned} \tag{3.2}$$

3.3 Modelo de Linguagem Mascarado

A principal contribuição da presente dissertação é introduzir a habilidade de transferência do conhecimento contido nas ricas representações contextualizadas bidirecionais fornecidas por um modelo de linguagem mascarado (MLM) para um modelo Seq2Seq. Como modelo de linguagem mascarado, foi selecionado o *ALBERT* [54], cuja arquitetura é similar ao popular modelo BERT [17], mas que alcança o estado da arte em diversas tarefas, apesar de possuir menos parâmetros que o BERT. Conforme apresentado na Seção 2.4, ALBERT se aproveita do compartilhamento de parâmetros para escalar durante o treinamento do modelo de linguagem e a inferência da tarefa alvo.

A partir do modelo de linguagem aprendido pelo ALBERT, obtém-se a distribuição de probabilidade dos tokens mascarados, dados os tokens não mascarados, de acordo com a Equação 3.3, onde ϕ são os parâmetros do ALBERT:

$$P_\phi(X^m|X^u) = P_\phi(x_1^m,\ldots,x_l^m|X^u) \tag{3.3}$$

onde $x_*^m \in X^m$ são os tokens mascarados, l indica o número de tokens mascarados em X e X^u são os tokens não mascarados.

Antes do treinamento do modelo principal, realizou-se um ajuste fino do ALBERT no conjunto de dados de treinamento disponível, usando somente o componente da função de custo que busca prever os tokens mascarados. Não se adotou o componente de *Sentence-order prediction*, pois, apesar de o ALBERT estar preparado para lidar com um par de sentenças como entradas, dada a definição do problema e do método de treinamento, apenas uma sentença é fornecida como entrada, tanto no ajuste do fino dos parâmetros dele quanto no treinamento do modelo principal, momento em que seus parâmetros não mudam. Como será visto na seção seguinte, MATTES usa a distribuição de probabilidade fornecida pelo ALBERT para cada token da sentença de entrada como rótulo para treinamento do modelo, em um dos componentes da função de custo. Assim, ao invés de forçar

o modelo a gerar uma distribuição de probabilidade com toda a massa de probabilidade em um só token, força-se o modelo a ter uma distribuição de probabilidade mais suave, injetando massa de probabilidade em diversos tokens.

3.4 Destilação de Conhecimento a partir do Modelo Mascarado

Nesta seção, será detalhado o componente de destilação de conhecimento da função de custo do modelo. Com o intuito de preservar o conteúdo da mensagem original, os atuais estudos do estado da arte na tarefa de aprendizado para TST utilizam uma técnica chamada de *back-translation* (BT) [53, 31, 15]. Proposta por [87] para a tarefa de tradução automática, trata-se de uma técnica para gerar sequências pseudo-paralelas para treinamento, quando não há sentenças paralelas disponíveis no conjunto de exemplos, gerando, assim, dados paralelos latentes. Dessa forma, no contexto da tarefa de TST, através da conversão automática das sentenças do conjunto de treinamento para outro estilo, criam-se pares de sentenças que são então treinados de maneira supervisionada.

Nessa linha, durante o treinamento, um modelo que faz uso de *back-translation* recebe uma sentença X e seu estilo s como entrada e converte-a para um outro estilo alvo $\hat{s} \neq s$, gerando a sentença $\hat{Y} = f_\theta(X, \hat{s})$. Após isso, a sentença gerada $\hat{Y}$ é repassada como entrada para o modelo juntamente com o estilo original s, e a rede é treinada para aprender a prever a sentença original de entrada X. Ou seja, primeiro a sentença de entrada é convertida para o estilo alvo, e depois ela é convertida de volta para seu estilo original.

O método proposto em [15] adota a estratégia de *back-translation* para aprender um modelo de TST, minimizando o valor negativo do logaritmo da probabilidade de a sentença gerada ser igual a sentença original, dada a sentença convertida, indicada por $f_\theta(X, \hat{s})$, e o estilo de entrada original s, conforme a Equação 3.4:

$$\mathcal{L}_{BT}(\theta) = -logP_\theta(Y = X | f_\theta(X, \hat{s}), s) \tag{3.4}$$

Durante o aprendizado, modelos Seq2Seq normalmente são treinados da esquerda para direita. Assim, durante o processo de geração de cada token que comporá a sentença, a distribuição de probabilidade sobre o vocabulário obtida será condicionada somente ao

tokens anteriores ao que está sendo previsto, para evitar que cada token veja ele próprio e os demais do futuro. Essa abordagem tem a desvantagem de que a distribuição de probabilidade obtida é estimada usando somente o contexto esquerdo.

Para superar essa limitação e gerar uma distribuição que contemple a informação bidirecional, durante o treinamento, MATTES usa uma configuração chamada de Transformer Mascarado. A adoção dessa arquitetura na tarefa de transferência de estilo originou-se da observação de que a distribuição de probabilidade de um token mascarado x_t^m dada por um MLM, por exemplo ALBERT, contém informações tanto do contexto passado quanto do futuro. Assim, como não se têm pares de sentenças disponíveis para realizar um treinamento supervisionado, a ideia central é forçar o MATTES a gerar uma distribuição conforme a fornecida pelo ALBERT para cada token. Dessa forma, garante-se que as sentenças geradas em tempo de inferência serão sempre provenientes da distribuição de probabilidade do estilo desejado. Espera-se que essa informação adicional tenha o potencial de melhorar a qualidade dos textos gerados e, em especial, o controle de estilo do modelo. Assim, durante a otimização do componente da função de custo referente ao *back-translation* (Equação 3.4), o alvo deixa de ser a distribuição em que toda massa de probabilidade está em um só token para ser a distribuição de probabilidade fornecida pelo modelo de linguagem mascarado $P_\phi(x_t^m|X^u)$ (Equação 3.3), para cada token x_t da sentença de entrada. A distribuição fornecida pelo modelo de linguagem mascarado se torna um alvo mais suave para a geração de texto durante o treinamento, fazendo com que o modelo se afaste de aprender uma distribuição mais abrupta e irreal, onde toda a massa de probabilidade está em um só token.

Outro ponto que o MATTES tenta alavancar ao usar um modelo de linguagem mascarado como o ALBERT é o uso de um esquema de destilação de conhecimento [33]. Em tal esquema, o treinamento de um modelo (chamado de *estudante*) se vale dos valores de saída de um outro modelo (chamado de *professor*) como objetivo, ao invés de usar rótulos oriundos do conjunto de treinamento [77]. Na prática, tem sido observado que a otimização e a regularização do processo de treinamento são mais bem controladas com o processo de destilação.

Assim, o MATTES se beneficia do esquema de destilação fazendo com que o modelo de linguagem mascarado assuma o papel de professor, enquanto que o modelo não-supervisionado Seq2seq se comporte como o estudante. A Equação 3.5 exibe como o componente de *back-translation* foi alterado para ser um componente bidirecional de destilação de conhecimento:

$$\mathcal{L}_{bidi}(\theta) = -\sum_{t=1}^{N}\sum_{w\in V} P_\phi(x_t = w|X^u)\cdot \\ \cdot log P_\theta(y_t = w|y_{1:t-1}, f_\theta(X, \hat{s}, s)) \tag{3.5}$$

onde $P_\phi(x_t)$ é o alvo suave fornecido pelo modelo de linguagem mascarado de parâmetros ϕ, N é o tamanho da sentença, e V denota o vocabulário. Pontua-se que os parâmetros do modelo de linguagem mascarado são fixos durante o processo de treinamento. A Figura 3.2 ilustra o processo de aprendizado, onde o objetivo é fazer com que a distribuição de probabilidade da palavra $P_\theta(y_t)$, fornecida pelo estudante, se aproxime da distribuição fornecida pelo professor, $P_\phi(x_t)$.

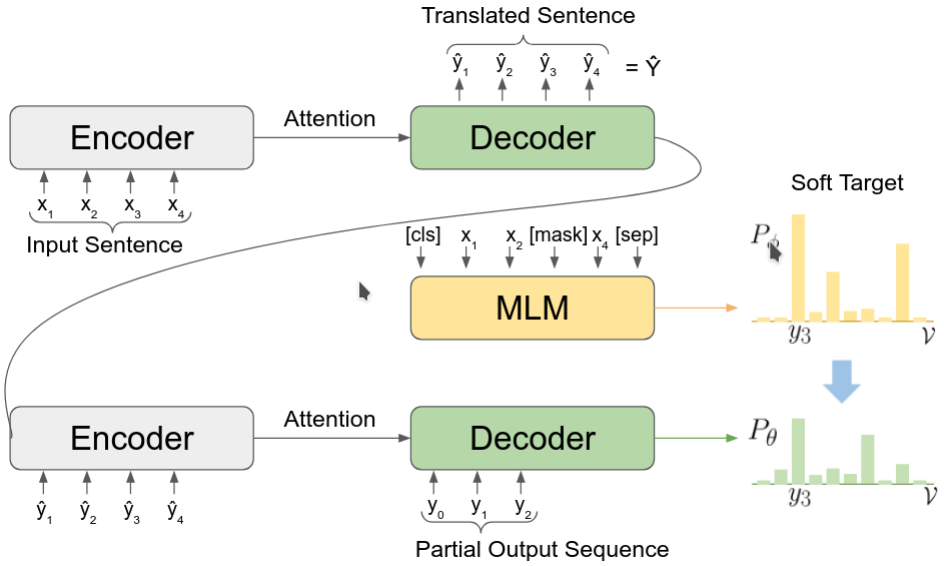

Figura 3.2: Ilustração do processo de treinamento durante a predição do token y_3

Essas estratégias de destilação e mascaramento foram inspiradas no estudo investigado em [12], no contexto da tarefa de tradução automática. Entretanto, enquanto o método proposto em [12] destila conhecimento de um modelo customizado de linguagem mascarado treinado a partir do modelo treinado seguindo a arquitetura do BERT [17], MATTES destila conhecimento do ALBERT [54], cuja arquitetura é similar a do BERT mas faz uso de compartilhamento de parâmetros, o que o torna um modelo menor e faz necessitar de menos recursos computacionais para seu ajuste fino e uso. Assim, MATTES é treinado com um componente de *back-translation* adaptado, conforme a Equação 3.6, chamado de *componente de destilação de conhecimento*:

$$\mathcal{L}_{KD}(\theta) = \alpha\mathcal{L}_{bidi}(\theta) + (1-\alpha)\mathcal{L}_{BT}(\theta) \tag{3.6}$$

onde α é um hiperparâmetro para ajustar a importância relativa dos alvos suaves fornecidos pelo modelo de linguagem mascarado e os alvos originais, onde toda massa de probabilidade é direcionada a um só token. O objetivo com essa abordagem é fazer com que a distribuição geradas pelo MATTES capture informação bidirecional, gerando uma distribuição mais realista e, consequentemente, melhore a qualidade da geração. Com a introdução do novo termo da função de custo, a distribuição é forçada a se tornar mais suave durante o treinamento.

Quando as tradicionais representações *one-hot* são usadas como alvo durante o treinamento, força-se o modelo a gerar o token correto. Todas os outros tokens não importam para o modelo, tratando igualmente tokens que seriam mais prováveis de ocorrer e tokens com quase zero chance de ocorrer. Dessa forma, confiando em uma distribuição mais suave, o modelo aumenta seu potencial de geração de sentenças mais fluentes, uma vez que a probabilidade se espalha sobre mais tokens. Além disso, uma distribuição mais suave evita a geração de tokens completamente fora do contexto e controla melhor o estilo desejado.

3.5 Algoritmo de Aprendizado

O treinamento do modelo aqui proposto segue o algoritmo definido em [15]. Ambas as redes, a discriminadora e a geradora, são treinadas de modo adversarial. Primeiramente, será descrito o aprendizado da rede discriminadora e, em seguida, o aprendizado da rede geradora, onde está inserido o *Transformer* Mascarado proposto neste trabalho.

3.5.1 Aprendizado da Rede Discriminadora

A rede neural discriminadora é basicamente um classificador multi-classe com $K+1$ classes. Neste caso, K classes estão associadas com K diferentes estilos e a classe restante refere-se às sentenças convertidas geradas pelo *Transformer* Mascarado. A rede discriminadora é treinada para distinguir entre as amostras originais de um estilo, amostras originais do outro estilo e as amostras artificiais geradas pela rede geradora. A função de custo do classificador multi-classes é dada pela Equação 3.7, onde ρ são os parâmetros da rede discriminadora:

$$\mathcal{L}_{discriminator}(\rho) = -logp_{\rho}(c|X) \tag{3.7}$$

onde c é o domínio estilístico da amostra e pode assumir $K + 1$ categorias, como falado anteriormente. Pontua-se que os parâmetros θ da rede geradora não são atualizados durante o treinamento da rede discriminadora.

3.5.2 Aprendizado da Rede Geradora

A função de custo final do *Transformer* Mascarado é composta de três componentes indicados pelas Equações 3.8, 3.9 e 3.10, descritas a seguir.

Componente de Reconstrução da Sentença de Entrada Quando o modelo recebe como entrada uma sentença X juntamente com seu estilo s, o modelo deve ser capaz de reconstruir a sentença original. Para que o modelo alcance essa habilidade, adicionou-se à função de custo um componente de reconstrução conforme a Equação 3.8:

$$\mathcal{L}_{self}(\theta) = -logP_\theta(Y = X|X, s) \tag{3.8}$$

Durante o treinamento, $\mathcal{L}_{self}$ é otimizada da maneira tradicional, ou seja, da esquerda para direita, mascarando-se o contexto futuro, conforme a Equação 3.2. Apesar de ser possível, não se adotou a técnica de destilação de conhecimento nesse componente, de forma a isolar a destilação a um componente só e verificar seu benefício. Ou seja, espera-se verificar se o MATTES consegue gerar a sentença X quando for lhe for fornecida como entrada a sentença X juntamente com seu estilo s, sem a necessidade de usar a técnica de destilação de conhecimento nesse ponto.

Componente de Destilação de Conhecimento Com o intuito de extrair conhecimento de um modelo de linguagem mascarado pré-treinado, nesse caso o ALBERT, e melhorar o processo transdutivo de converter uma sentença de um domínio estilístico para outro, adicionou-se um componente de destilação de conhecimento que segue a Equação 3.9, com $\mathcal{L}_{BT}$ definido na Equação 3.4 e $\mathcal{L}_{bidi}$ definido pela Equação 3.5:

$$\mathcal{L}_{KD}(\theta) = \alpha\mathcal{L}_{bidi}(\theta) + (1 - \alpha)\mathcal{L}_{BT}(\theta) \tag{3.9}$$

Com isso, espera-se suavizar a distribuição de probabilidade do modelo conversor de

estilo, produzindo sentenças mais fluentes que se assemelham aos textos do domínio alvo.

Componente Adversarial Se o modelo for treinado somente com os componentes de reconstrução da sentença e de destilação de conhecimento, que tentam otimizar as funções de custo $\mathcal{L}_{self}(\theta)$ e $\mathcal{L}_{KD}(\theta)$, respectivamente, o modelo poderia convergir rapidamente para apenas aprender a copiar a sentença de entrada, ou seja, se ater a aprender a função identidade. Assim, para evitar esse comportamento problemático e indesejado, um componente adversarial é adicionado na função de custo para encorajar que os textos convertidos para o estilo $\hat{s}$, diferente do estilo da sentença de entrada s, se aproximem dos textos que sigam o estilo $\hat{s}$. A sentença convertida $\hat{Y}$ é inserida na rede neural discriminadora e, durante o treinamento, maximiza-se a probabilidade de a sentença gerada ser do estilo $\hat{s}$, tentando otimizar a função de custo definida na Equação 3.10. Pontua-se que os parâmetros ρ da rede discriminadora não atualizados durante o treinamento da rede geradora.

$$\mathcal{L}_{adversarial}(\rho) = -logp_{\rho}(c = \hat{s}|f_{\theta}(X, \hat{s})) \tag{3.10}$$

3.5.3 Treinamento Adversarial Geral

Redes adversariais geradoras ou GANs [26] são redes neurais geradoras diferenciáveis e baseadas em um cenário de teoria dos jogos onde uma rede geradora deve competir contra um adversário. A rede geradora produz amostras $x = g(z; \theta^{(g)})$. Seu adversário, a rede discriminadora, busca distinguir entre amostras do conjunto de treinamento das amostras provenientes da rede geradora. Dessa forma, durante o treinamento, o discriminador tenta aprender a classificar corretamente as amostras como reais ou artificialmente geradas pela rede geradora. Simultaneamente, a rede geradora tenta enganar o discriminador e produzir amostras que se pareçam com amostras provenientes da distribuição de probabilidade do do conjunto de dados. Na convergência, as amostras do gerador serão indistinguíveis das amostras reais do conjunto de treinamento e a rede descriminadora poderá ser descartada.

No contexto dessa dissertação, a adoção de um treinamento adversarial é motivada pelo possibilidade de converter textos para o estilo desejado sem incorrer na falha de copiar o texto de entrada. O procedimento geral de treinamento consiste em, repetidamente, realizar n_d passos de treinamento do discriminador, minimizando $\mathcal{L}_{discriminator}(\rho)$, seguidos

de n_f passos de treinamento da rede geradora, minimizando $a_1\mathcal{L}_{self}(\theta) + a_2\mathcal{L}_{KD}(\theta) + \mathcal{L}_{adversarial}(\rho)$, até a convergência. a_1 e a_2 são hiperparâmetros do modelo que servem para ajustar a importância de cada um dos componentes na função de custo da rede geradora. Durante os passos de treinamento do discriminador, somente os parâmetros ρ da rede discriminadora são atualizados. De forma análoga, durante os passos de treinamento da rede geradora, só os parâmetros θ do *Transformer* Mascarado são atualizados. A Figura 3.3 ilustra o treinamento adversarial adotado por MATTES.

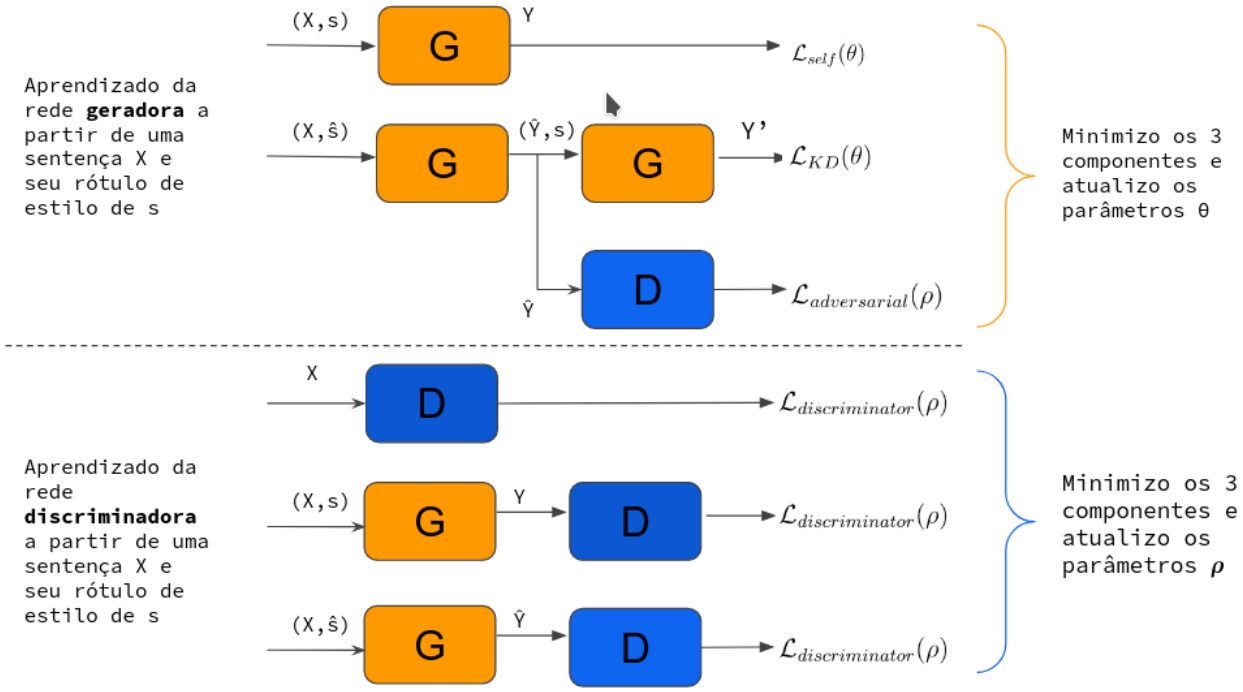

Figura 3.3: Ilustração do treinamento adversarial adotado. G indica a rede geradora e D a rede discriminadora

Destaca-se que, durante o processo de *back-translation* do treinamento (Figura 3.2), não geramos a sentença transformada para o outro domínio $\hat{Y}$ por meio de amostragem ou de decodificação gulosa da distribuição de probabilidade transdutiva fornecida pela rede geradora. Essa decisão decorre do fato de que propagar os gradientes para trás, passando por operações estocásticas discretas, não é diferenciável, e tipicamente envolve técnicas como REINFORCE [102] ou o artifício de aproximação da distribuição de Gumbel-Softmax [43], os quais sofrem de alta variância [31]. Para superar essa dificuldade, a distribuição *softmax* gerada pelo decodificador do MATTES é inserida como entrada novamente para o codificador do MATTES, juntamente com o estilo da sentença original, conforme [15]. Nessa configuração, é possível propagar diretamente os gradientes a partir das redes discriminadora e geradora até o modelo reverso que gerou as sentenças traduzidas para o estilo alvo.

Relembrando as questões 1 e 2 de pesquisa enunciadas no capítulo 1, tem-se: *Como modelos de linguagem mascarados podem ser utilizados para melhorar o desempenho de modelos neurais na tarefa de transferência de estilo e consequentemente gerar textos de alta qualidade?* e *Como contornar as dificuldades que modelos de linguagem mascarados enfrentam para gerar textos quando eles são inseridos no processo de transferência de estilo?* Buscando respondê-las, esse capítulo propôs um método de aprendizado de máquina que aborda a tarefa de TST não-supervisionada usando um modelo de linguagem mascarado no treinamento, inserindo-o em uma configuração de destilação de conhecimento, para extrair conhecimento dele e enriquecer a qualidade da geração dos textos.

Capítulo 4

Resultados Experimentais

Esse capítulo apresenta a metodologia experimental adotada para a realização dos experimentos (Seção 4.1) e os resultados alcançados (Seção 4.2). Assim, o capítulo inclui os conjuntos de dados referentes às tarefas de transferência de estilo usadas para avaliar o MATTES, os modelos cujos resultados foram comparados com os do MATTES, as métricas de avaliação usadas e detalhes do treinamento do modelo, e os resultados quantitativos computados a partir de cinco métricas, os resultados das avaliações com pessoas e comparações com trabalhos relacionados.

4.1 Metodologia Experimental

Nesta seção, os experimentos realizados juntamente com suas respectivas configurações de treinamento serão pormenorizados.

4.1.1 Conjunto de Dados e Tarefas

MATTES foi avaliado em duas tarefas de transferência de estilo na língua inglesa: imitação autoral e transferência de sentimento.

A tarefa de **imitação autoral** consiste em converter o estilo de uma sentença para o estilo de um determinado autor. Almeja-se gerar uma sentença que seja uma paráfrase da sentença original, mas que possua um estilo textual diferente. Para verificar a habilidade do MATTES executar essa tarefa, foi usado um conjunto de 21.000 (vinte e uma mil) sentenças de peças do autor inglês William Shakespeare, transcritas para o inglês moderno. Como são traduções do inglês para o inglês, esse material também pode ser considerando como tradução intra-linguística. A tarefa de imitação autoral pode ser pensada como

uma adaptação, onde se modifica um texto dentro da própria língua. Em especial, tal tarefa pode ser pensada como a soma de dois processos: uma adaptação temporal e outra linguística. Esse conjunto de dados foi curado em [108] e usado previamente em trabalhos de transferência de estilo textual não-supervisionada [31, 50]. Ele foi dividido em conjunto de treino, de validação e de testes. Adotou-se s_1 para denotar o inglês *moderno* e $\mathcal{D}_1$ para o conjunto de dados referente a esse estilo. Ainda, adotou-se s_2 para denotar o inglês *Shakesperiano* e $\mathcal{D}_2$ para denotar o domínio das sentenças em inglês *Shakesperiano*. MATTES é avaliado tanto para transferir sentenças de s_1 para s_2 como de s_2 para s_1. Embora existam os pares de sentença, eles não são usados de forma pareada durante o treinamento, mas apenas para avaliar os resultados.

A tarefa de **Transferência de Sentimento** consiste em converter a sentença para um sentimento diferente, preservando seu conteúdo semântico principal. Nessa tarefa, foi usado o conjunto de dados YELP, coletado em [88], que inclui avaliações de estabelecimentos realizadas dentro do aplicativo YELP. O conjunto de dados contém 250.000 sentenças negativas e 380.000 sentenças positivas. Para avaliar de forma quantitativa as habilidades de generalização dos modelos de transferência, como conjunto de testes, foram usadas 1.000 sentenças paralelas anotadas por humanos, introduzidas em [56]. O estilo de sentimento positivo é denotado por s_1 e seu domínio é $\mathcal{D}_1$, enquanto que o estilo de sentimento negativo é denotado por s_2 e seu estilo por $\mathcal{D}_2$. Novamente, MATTES é avaliado tanto para transferir sentenças de s_1 para s_2 como de s_2 para s_1.

4.1.2 Baselines

MATTES foi comparado com modelos recentes que possuem as sentenças do conjunto de testes convertidas disponíveis. Na tarefa de imitação autoral, os resultados foram comparados com os modelos *Deep Latent Sequence* (DLSM) [31] e STRAP [50], os quais demonstraram resultados relevantes nas métricas de preservação de conteúdo. Como o modelo *Style Transformer* [15] não realizou essa tarefa, foi feita uma implementação própria do seu modelo e geraram-se amostras a partir dele para que ele também pudesse ser comparado com MATTES. Com relação à tarefa de transferência de sentimento, os resultados do MATTES foram comparados com os resultados de outros modelos que obtiveram o estado da arte para essa tarefa, incluindo novamente o DLSM [31] e *Style Transformer* [15], além do modelo proposto em [88] e ainda os modelos RETRIEVE-ONLY, RULE-BASED e DELETEANDRETRIVE, que obtiveram os melhores resultados de acordo com os experimentos divulgados em [56].

4.1.3 Avaliação Quantitativa

Criar métricas automáticas de avaliação de textos gerados artificialmente que correlacionam bem com o julgamento humano é ainda um campo aberto de pesquisa. Porém, avaliações com métricas automáticas são muito mais baratas e rápidas de serem executadas, quando comparadas com avaliações humanas. Dessa forma, primeiramente foram selecionadas métricas de avaliação automática vastamente usadas na literatura [110, 53, 31, 50] sempre procurando focar nas três dimensões que se entende que sistemas efetivos de transferência de estilo textual devem possuir, a saber, controle de estilo, preservação de conteúdo e fluência.

Com relação ao **controle de estilo**, empregou-se um classificador neural convolucional [48] para medir quão bem MATTES consegue controlar o estilo dos textos gerados. Para tanto, dois classificadores são treinados, um para cada tarefa, objetivando acertar a qual domínio estilístico uma sentença pertence. A métrica de controle de estilo, para cada tarefa, é a acurácia que esse classificador confere às sentenças geradas pelo MATTES.

Considerando que a **preservação de conteúdo** é a característica mais desejada do modelo, três métricas distintas foram usadas para avaliar os modelos de acordo com essa dimensão: BLEU [72], BertScore [113] e similaridade semântica (SIM) [99]. Os valores das três métricas são calculados usando a sentença gerada pelo modelo e uma sentença de referência previamente anotada.

BLEU (*Bilingual Evaluation Understudy Score*) é uma métrica rápida e barata amplamente usada em tarefas de PLN, proposta inicialmente para tradução automática, e que pode ser usada em diversas línguas e correlaciona bem com julgamentos humanos. Esclarece-se que, apesar de o BLEU ser tradicionalmente usado para tradução inter-linguística, ele foi adotado nesse estudo para avaliar as adaptações dos textos de Shakespeare, que consistem em uma tarefa de tradução intra-linguística. Nos experimentos, BLEU foi calculado usando o pacote NLTK [5]. BLEU é calculado conforme as Equações 4.1, onde c é o comprimento da sentença candidata traduzida, r é o comprimento da sentença de referência, BP é um termo que penaliza a diferença entre os comprimento das sentenças candidata e de referência, k é o máximo n-gram que se quer avaliar e p_n é a pontuação de precisão para os grams de comprimento n:

$$
\begin{aligned}
p_n &= \frac{\#\text{ n-grams coincidentes}}{\#\text{ n-grams na sentença candidata}} \\
BP &= e^{min(0,1-\frac{r}{c})} \\
BLEU &= BP \times \prod_{i=1}^{k} p_n^{w_n}
\end{aligned}
\tag{4.1}
$$

Segundo os valores propostos em [72], foram considerados os valores de $k = 4$ e pesos uniformes $w_n = 1/N$.

O F_1 BERTscore [113] calcula a pontuação de similaridade para cada token da sentença candidata com cada token da sentença de referência. Diferente da métrica BLEU, ao invés de correspondências exatas, a similaridade entre tokens é computada usando representações vetoriais contextualizadas fornecidas por uma modelo de linguagem mascarado, no caso o BERT[17]. Para uma sentença tokenizada de referência $X = (x_1, \ldots, x_k)$, BERT gera uma lista de vetores para cada token $(\mathbf{x_1}, \ldots, \mathbf{x_k})$. De maneira análoga, para uma sentença candidata $\hat{X} = (\hat{x_1}, \ldots, \hat{x_m})$, BERT gera uma lista de vetores para cada token $(\mathbf{\hat{x_1}}, \ldots, \mathbf{\hat{x_m}})$. As pontuações de revocação, precisão e F_1 são calculadas conforme as Equações 4.2:

$$
\begin{aligned}
R_{bert} &= \frac{1}{|x|} \sum_{x_i \in x} \max_{\hat{x_j} \in \hat{x}} \mathbf{x_i^T \hat{x}_j} \\
P_{bert} &= \frac{1}{|\hat{x}|} \sum_{\hat{x_j} \in \hat{x}} \max_{x_i \in x} \mathbf{x_i^T \hat{x}_j} \\
F_{bert} &= 2\frac{P_{bert} \times R_{bert}}{P_{bert} + R_{bert}}
\end{aligned}
\tag{4.2}
$$

Buscando superar as limitações da métrica BLEU, bem como evitar dar crédito parcial e penalizar candidatas semanticamente corretas quando diferem lexicalmente da sentença de referência, em [99] foi introduzida uma métrica chamada SIM para medir a similaridade semântica entre sentenças. Para tanto, foi treinado um modelo codificador g que tenta maximizar a similaridade de pares de sentenças presentes em um conjunto de dados de paráfrases [100]. O codificador g calcula a média das representações vetoriais da cada token de uma sentença para criar uma representação vetorial da sentença. A similaridade entre um par de sentenças $< X, \hat{X} >$, SIM, é obtida pela codificação das duas sentenças por g e depois pelo cálculo da similaridade de cossenos das duas representações, conforme

a Equação 4.3:

$$SIM = cos(g(X), g(\hat{X})) \tag{4.3}$$

Por fim, mediu-se também a **fluência** da sentença gerada por meio do cálculo da perplexidade (PPL) das sentenças geradas a partir de modelos de linguagem com redes recorrentes do tipo Longo Short-Term Memory (LSTM) [36], treinados em cada domínio. Seja J o erro de entropia cruzada médio em um corpus de tamanho T. Tem-se que a perplexidade é dada pela Equação 4.4. Quanto menor ela for, mais fluente é o texto:

$$\begin{aligned} J &= \frac{1}{T}\sum_{t=1}^{T} J^{(t)}(\theta) \\ Perplexidade &= 2^J \end{aligned} \tag{4.4}$$

4.1.4 Avaliação com Pessoas no Conjunto de Testes

Como métricas automáticas de avaliação só fornecem uma aproximação superficial da qualidade das sentenças convertidas, avaliações com pessoas também foram conduzidas sobre as saídas dos modelos. Para limitar a quantidade de modelos avaliados, foram estabelecidos valores mínimos para as métricas de controle de estilo e de preservação de conteúdo BLEU, e somente modelos acima desses valores mínimos foram submetidos a avaliações com pessoas, seguindo a abordagem descrita em [53]. Voluntários avaliaram esses modelos no mesmo conjunto de validação e o melhor modelo foi executado no conjunto de testes. Os voluntários anônimos tiveram acesso ao Termo de Consentimento Livre e Esclarecido (TCLE) e concordaram em nos ajudar a avaliar as sentenças, considerando que o tempo gasto na tarefa seria de aproximadamente 15 minutos. O processo foi todo conduzido por meio de formulários online e nenhum contato pessoal foi realizado durante a avaliação, para evitar influências e vieses. As avaliações foram coletadas entre os dias 12/05/2021 e 17/05/2021 e o perfil dos voluntários era de nível superior e fluente na leitura da Língua Inglesa. Um exemplo de formulário de avaliação com o TCLE encontra-se no Apêndice A.3.

Assim, para complementar e superar as limitações de métricas automáticas, avaliações humanas foram realizadas no conjunto de dados do YELP via *crowd-sourcing*. Embora em um primeiro momento o conjunto de dados do Shakespeare também tenha sido repassado para a avaliação com pessoas, os resultados foram bastante inconclusivos, uma vez que o domínio da língua inglesa mais próxima da arcaica é muito complexo para fa-

lantes não-nativos de inglês. Dessa forma, aleatoriamente, selecionaram-se sessenta (60) sentenças do conjunto de testes do YELP somente (30 de cada sentimento) para avaliação humana. As sentenças geradas pelo MATTES foram comparadas com as geradas pelos modelos propostos em [31] e em [15], visto que ambos possuem amostras publicamente disponíveis. Os respondentes recebiam uma sentença de entrada e três sentenças anônimas, para responder às seguintes questões em inglês:

- *Which sentence has the most positive/negative sentiment toward the input sentence?*
- *Which sentence retains most content from the input sentence?*
- *Which sentence is the most fluent one?*

Também havia uma opção "No Preference" para o respondente, para o caso de as sentenças serem igualmente boas ou ruins.

4.1.5 Hiperparâmetros e Detalhes do Treinamento

Durante os experimentos, algumas combinações de hiperparâmetros foram avaliadas de acordo com o desempenho do MATTES no conjunto de validação. Variou-se o termo α do componente de destilação de conhecimento da função de custo (Equação 3.9) com os valores $\{0.1, 0.5\}$. A temperatura T_{KD} de destilação de conhecimento também foi variada em $\{1, 5, 10\}$. Testaram-se também algumas configurações para o número de passos do discriminador n_d e o número de passos do gerador n_f executados durante o treinamento. Variou-se a tupla $(n_d,\ n_f)$ com os valores $\{(7, 5), (9, 5), (10, 5)\}$. Variou-se também a contribuição de cada componente (Equações 3.8, 3.9, and 3.10) na função de custo final do MATTES ($a_1\mathcal{L}_{self}(\theta)+a_2\mathcal{L}_{KD}(\theta)+\mathcal{L}_{adversarial}(\rho)$). Como [15], concluiu-se que executar *dropout* aleatório nos tokens da sentença de entrada durante o cálculo do componente de reconstrução da sentença impacta positivamente no resultados do modelo. Variou-se a taxa de *dropout* em $\{0.2, 0.3, 0.4\}$.

Ressalta-se que todos os experimentos foram realizados usando a linguagem de programação Python, usando o arcabouço de programação PyTorch. A implementação do ALBERT usa o arcabouço *HuggingFace transformers* [103] que é baseado em PyTorch.

Antes do treinamento principal do MATTES, para cada tarefa executada, escolheu-se o modelo disponível ALBERT-LARGE-V2 e realizou-se um ajuste fino de dois ALBERTs desse tipo, um para cada domínio estilístico, usando somente o objetivo tradicional do

modelo de linguagem de mascarado. Nesse pré-treinamento adaptativo de domínio, como em [30], treinou-se cada ALBERT por 100 épocas, usou-se o otimizador AdamW com ϵ igual a $1e-6$, taxa de aprendizado linear com *warm-up* e taxa máxima de aprendizado de $1e-5$. ALBERT não usa *dropout* nem regularização em seu treinamento. O treinamento ocorreu em 4 GPUs NVIDIA P100 com um lote de treinamento de 8 exemplos por GPU. Após o ajuste fino do ALBERT em cada domínio estilístico, extraíram-se os *logits*[1] para cada token de cada sentença do conjunto de treinamento. Seguindo [12], para otimização dos recursos computacionais, foram considerados apenas os top-8 *logits* para serem usados como rótulos durante o treinamento principal de MATTES.

Todos os experimentos foram executados em GPUs Tesla P100-SXM2 usando o sistema operacional Ubuntu, em uma máquina com processador Intel(R) Xeon(R) CPU E5-2698 v4. Na média, o conjunto de dados do Yelp leva 30 horas com um lote de treinamento de 192, enquanto o conjunto de dados do Shakespeare leva 15 horas com um lote de 64.

O esquema arquitetural do MATTES inclui uma arquitetura *Transformer* de quatro camadas com quatro cabeças de atenção em cada camada, tanto para o codificador, quanto para o decodificador e a rede discriminadora, conforme [15]. As representações vetoriais dos tokens, dos estados ocultos e da posição do token na sentença possuem 256 dimensões. Outros dois vetores de dimensão 256 são incluídos para representar os domínios estilísticos. O vetor do estilo alvo é inserido no codificador como um token adicional da sentença.

Em todas as sequências do conjunto de dados, no processo de tokenização não foi inserido o token de início de sentença, somente o token de fim de sequência. Durante o treinamento, ao injetarmos as sequências na rede neural Transformer, usamos um token de estilo como sendo o primeiro token da sequência. Assim, os vetores que representam os estilos também são treinados e fazem parte do modelo.

Os hiperparâmetros dos melhores modelos em cada uma das tarefas executadas estão listados no Apêndice A.1.

4.2 Resultados

Os resultados apresentados consideram os valores obtidos de acordo com as métricas quantitativas, as avaliações com pessoas e estudos ablativos.

[1] São as representações não normalizadas fornecidas pelo ALBERT antes de passarem pela camada Softmax.

4.2.1 Resultados da Avaliação Quantitativa

Tabela 4.1: Resultados com métricas automáticas de avaliação nas tarefas de transferência de sentimento e imitação autoral. Mostramos também a PPL dada pelos LM nos conjuntos de testes de ambos os domínios.

Tarefa	Modelo	Acc.	BLEU	F_1	SIM	PPL_{D_1}	PPL_{D_2}
Tranferência de Sentimento	Conjunto de Testes	-	-	-	-	19.16	24.93
	CAE [88]	74.2	4.57	.88	-	44.49	43.09
	Retrieval [56]	**94.3**	0.72	.86	-	**25.39**	**25.12**
	Rule-based [56]	85.6	15.57	.90	-	109.10	135.09
	DeleteAndRetrieve [56]	89.1	12.05	.89	-	34.63	71.53
	Deep Latent Seq.[31]	84.4	16.59	.91	48.8	33.16	25.40
	Style Transformer [15]	84.6	**21.86**	**.92**	**60.9**	73.74	46.36
	MATTES (ours)	88.3	20.52	**.92**	59.2	69.33	57.70
Imitação autoral	Conjunto de Testes	-	-	-	-	64.57	100.34
	Deep Latent Seq. [31]	**80.4**	9.04	.89	43.4	**33.74**	**37.78**
	STRAP [50]	70.0	8.27	**.90**	**56.4**	40.44	53.27
	Style Transf.	61.4	11.30	.89	52.8	102.31	79.26
	MATTES (ours)	68.9	**11.73**	.89	54.0	89.99	91.60

A Tabela 4.1 exibe os resultados usando as métricas automáticas. No conjunto de dados do YELP, compararam-se os melhores modelos exibidos em [56] e complementou-se com [88], [31], e [15], com exceção da métrica SIM que não foi calculada para os trabalhos de [56] e [88], uma vez que isso demandaria tempo específico para esses modelos. No conjunto de dados de Shakespeare, comparou-se MATTES com os resultados de [31] e [50]. Além disso, os resultados também foram comparados com nossa própria implementação do modelo *Style Transformer* [15] na tarefa de imitação autoral, uma vez que o trabalho original não incluiu essa tarefa. Pontua-se que essa implementação é parecida com MATTES, mas usa o componente tradicional de *back-translation* na função de custo, ao invés do componente de destilação de conhecimento.

No conjunto de dados de Shakespeare, MATTES superou todos as abordagens anteriores com relação a métrica de preservação de conteúdo BLEU. Com relação às duas outras métricas de preservação de contéudo, MATTES ficou atrás somente do modelo STRAP [50]. Esses resultados indicam que MATTES tem a habilidade de preservação do conteúdo léxico, enquanto STRAP é mais forte na preservação do conteúdo semântico. Entretanto, esses resultados acabam sendo consequências naturais da modelagem dos métodos. STRAP treina um modelo Seq2Seq a partir de um modelo de linguagem autorregressivo pré-treinado GPT-2 [81] em um conjunto de dados de paráfrases [100]. A partir desse modelo, é possível criar pares de sentenças (X, Y) a partir das sentenças do conjuntos de dados de Shakespeare. A partir desses pares de sentença, STRAP é treinado de

maneira supervisionada, partindo inicialmente também do modelo GPT2. Como STRAP é pré-treinado em uma enorme quantidade de paráfrases, é natural que a similaridade semântica dele seja superior, visto que a paráfrase de uma sentença é uma outra sentença com mesmo significado semântico. Outra explicação para a métrica SIM do STRAP ser superior ao valor obtido por MATTES reside no fato de o modelo que calcula a métrica SIM ter sido treinado no mesmo conjunto de dados [100] que STRAP treina para gerar as paráfrases que tornarão possível o treinamento supervisionado. Dessa forma, STRAP foi treinado indiretamente para gerar sentenças com a métrica SIM elevada.

Ainda na tarefa de imitação autoral, o método *Deep Lantent Sequence Model* (DLSM) [31] possui a melhor métrica de controle de estilo e de fluência, porém, às custas de valores nas métricas de preservação de conteúdo bem inferiores. Nos estudos ablativos da Subseção 4.2.3, mostra-se que MATTES supera DLSM tanto no controle de estilo quanto na preservação do conteúdo, por meio de ajustes nos hiperparâmetros do modelo. Além disso, DLSM muitas vezes produz sentenças triviais que tem baixa perplexidade, mas com muita perda de conteúdo, que será visto também nos estudos ablativos.

MATTES alcançou um equilíbrio de acurácia e BLEU no conjunto de dados do YELP, se aproximando bastante de todas as melhores métricas. Dentre os que obtiveram as melhores métricas de preservação de conteúdo, MATTES obteve o melhor controle de estilo, o que indica que a adoção dos alvos suaves provenientes do ALBERT durante o treinamento melhorou as distribuições de conversão, gerando amostras que se assemelham a amostras provenientes da distribuição do estilo desejado, impulsionando as capacidades de generalização do processo de transferência de estilo.

4.2.1.1 Testes Estatísticos

Foram realizados testes estatísticos para comparar as métricas BLEU e SIM obtidas pelo MATTES e pelos outros dois modelos que obtiveram as melhores métricas de preservação de conteúdo. Denota-se H_0 como a hipótese nula e H_1 como a hipótese alternativa a ser testada. Adotou-se o nível de significância $\alpha = 0,05$ e assumiu-se que as variáveis aleatórias $\bar{B}$ e $\bar{S}$, que correspodem, respectivamente, à média populacional do BLEU e do SIM, possuem distribuição normal para cada modelo. Os testes foram baseados nas médias amostrais das duas métricas no conjunto de testes. O valor-p é a probabilidade de se obter uma estatística de teste igual ou mais extrema que aquela observada em uma amostra, sob a hipótese nula. Assim, quando se obtém um valor-p menor que o nível de significância, isso indica um evento extremamente improvável, indicando rejeição de H_0.

Tabela 4.2: Testes t da média populacional do BLEU, para cada modelo e tarefa.

Tarefa	**Teste t**	H_0	H_1	Resultado	valor-p
Tranferência de Sentimento	MATTES-DLSM	$\bar{B}_{MATTES} = \bar{B}_{DLSM}$	$\boldsymbol{\bar{B}_{MATTES} > \bar{B}_{DLSM}}$	Rejeito H_0	$p < 10^{-3}$
	MATTES-STYLE	$\boldsymbol{\bar{B}_{MATTES} = \bar{B}_{STYLE}}$	$\bar{B}_{MATTES} < \bar{B}_{STYLE}$	Aceito H_0	$p = 0.13$
Imitação autoral	MATTES-STRAP	$\bar{B}_{MATTES} = \bar{B}_{STRAP}$	$\boldsymbol{\bar{B}_{MATTES} > \bar{B}_{STRAP}}$	Rejeito H_0	$p < 10^{-3}$
	MATTES-STYLE	$\boldsymbol{\bar{B}_{MATTES} = \bar{B}_{STYLE}}$	$\bar{B}_{MATTES} > \bar{B}_{STYLE}$	Aceito H_0	$p = 0.25$
Tranferência de Sentimento	MATTES-DLSM	$\bar{S}_{MATTES} = \bar{S}_{DLSM}$	$\boldsymbol{\bar{S}_{MATTES} > \bar{S}_{DLSM}}$	Rejeito H_0	$p < 10^{-3}$
	MATTES-STYLE	$\bar{S}_{MATTES} = \bar{S}_{STYLE}$	$\boldsymbol{\bar{S}_{MATTES} < \bar{S}_{STYLE}}$	Rejeito H_0	$p = 0.05$
Imitação autoral	MATTES-STRAP	$\bar{S}_{MATTES} = \bar{S}_{STRAP}$	$\boldsymbol{\bar{S}_{MATTES} < \bar{S}_{STRAP}}$	Rejeito H_0	$p < 10^{-3}$
	MATTES-STYLE	$\bar{S}_{MATTES} = \bar{S}_{STYLE}$	$\boldsymbol{\bar{S}_{MATTES} > \bar{S}_{STYLE}}$	Rejeito H_0	$p = 0.03$

Os resultados dos testes estatísticos da tabela 4.2 reforçaram os resultados da tabela 4.1. Com relação à métrica BLEU, MATTES superou os demais modelos baselines, com exceção do modelo *Style Transformer*, onde estatisticamente, para um intervalo de confiança de 95%, não é possível afirmar a superioridade de nenhum modelo. Com relação à métrica de similaridade de semântica, o modelo STRAP é estatisticamente superior ao MATTES na tarefa de imitação autoral. Na tarefa de transferência de sentimento, o *Style Transformer* conseguiu superar o MATTES no limite do intervalo de confiança adotado. Nas comparações restantes, MATTES é superior estatisticamente como se observa.

4.2.2 Resultados da Avaliação com Pessoas

Também foram conduzidas avaliações humanas devido às já comentadas limitações nas métricas automáticas de avaliação. A Tabela 4.3 mostra a pontuação obtida por MATTES comparada com os modelos *Style Transformer* e DLSM, em cada uma das três dimensões avaliadas.

Tabela 4.3: Resultados das avaliações humanas no conjunto de dados do YELP. Quando modelos diferentes geram a mesma sentença, uma resposta pode pontuar mais de um modelo.

Modelo	**Sentimento**	**Conteúdo**	**Fluência**
DLSM	28.0	25.8	**38.6**
Style Tranformer	28.0	19.5	22
MATTES	**32.6**	22.5	30.1
No Prefer.	18.6	**37.3**	17.4

MATTES alcançou o maior resultado na métrica de controle de estilo, enquanto que o DLSM ficou com a melhor fluência, na linha dos resultados das métricas automáticas. A opção "No Preference" foi a mais escolhida pelo avaliadores com relação à métrica de preservação de conteúdo, o que indica que os modelos geraram sentenças igualmente boas

ou ruins. Nessa linha, destaca-se que houve uma sentença, dentre as 60 aleatoriamente escolhidas a partir do conjunto de testes, em que os três modelos geraram exatamente a mesma sentença. Também se associou esse resultado ao fato de que, na tarefa de transferência de sentimento, altera-se de certo modo o conteúdo em algum grau, fazendo com que a questão referente à preservação de conteúdo seja um pouco enganosa para os respondentes, os quais são primeiramente não-nativos da língua inglesa.

4.2.3 Estudos Ablativos

Nesta seção, busca-se mensurar o impacto individual da técnica de destilação de conhecimento no desempenho geral do MATTES e realizar uma análise qualitativa entre as amostras geradas pelo MATTES e pelo DLSM.

4.2.3.1 Componente de Destilação de Conhecimento

Com intuito de mensurar o impacto causado pelo componente de destilação de conhecimento da função de custo, $\mathcal{L}_{KD}(\theta)$, conduziram-se experimentos adicionais em ambas as tarefas. A versão implementada do *Style Tranformer* [15] foi treinada e comparada com algumas versões do MATTES. Os dois modelos foram treinados exatamente com os mesmos hiperparâmetros, exceto aqueles que se aplicam somente ao componente de destilação de conhecimento da função de custo, os quais são Temperatura T e α. A Tabela 4.4 exibe os resultados.

Tabela 4.4: Estudos de ablação do componente de Destilação de Conhecimento.

Conjunto de Dados	**Modelo**	**Acc**	**BLEU**	$\boldsymbol{F_1}$
Yelp	Style Tranf.	86.2	19.04	.91
Yelp	$T = 5$ $\alpha = 0.5$	**89.9**	19.16	.92
Yelp	$T = 10$ $\alpha = 0.5$	88.3	**20.52**	**.92**
Shakespeare	Style Tranf	49.8	11.17	.89
Shakespeare	$T = 5$ $\alpha = 0.5$	**81.3**	9.13	.89
Shakespeare	$T = 1$ $\alpha = 0.5$	72.0	11.24	.89
Shakespeare	$T = 5$ $\alpha = 0.1$	65.7	**11.31**	**.89**

Os resultados indicam que a substituição do componente tradicional de *back-translation* pelo componente de destilação de componente proposto nessa dissertação melhora a qualidade do modelo Seq2seq. Resultados também mostram que é possível controlar o balanceamento entre as métricas de controle de estilo e de preservação de conteúdo (BLEU) baseado nas diferentes escolhas dos hiperparâmetros. Por exemplo, elevando-se o valor

mínimo estabelecido para acurácia, MATTES supera [31], tanto na métrica de controle de estilo quanto na de preservação de conteúdo.

4.2.3.2 Comparação entre MATTES e DLSM

Realizou-se uma análise mais qualitativa entre MATTES e *Deep Latent Sequence Model* (DLSM) por meio da análise das sentenças convertidas na tarefa de imitação autoral. Essa tarefa foi escolhida pois as conversões das sentenças do conjunto de testes para o DLSM estão disponíveis. Dessa análise, observou-se que ambos os modelos são capazes de gerar boas conversões entre estilos, porém MATTES tende a preservar melhor o conteúdo. Além disso, DLSM tende a criar sentenças muito curtas quando a sentença de entrada é longa, perdendo conteúdo. Essa observação é corroborada pelo fato de o valor da métrica de preservação de conteúdo do MATTES ser a maior. Acredita-se que os mecanismos de atenção presentes no MATTES podem ter ajudado a lidar com sentenças longas. As sentenças geradas pelo DLSM também possuem perplexidades inferiores ao próprio conjunto de testes, indicando sentenças pequenas ou triviais. A Tabela 4.5 exibe alguns exemplos de conversões de sentenças longas.

Tabela 4.5: Sentenças transferidas na tarefa de imitação autoral

Model	Shakespeare to Modern
Source	Lo, here upon thy cheek the stain doth sit Of an old tear that is not washed off yet.
Ref.	There's still a stain on your cheek from an old tear that hasn't been washed off yet.
DLSM	Right on, now.
MATTES	actually, here on your face the stains up i do sit of an old tears that is not washed off yet.
Source	These happy masks that kiss fair ladies'brows, Being black, puts us in mind they hide the fair.
Ref.	Look at other beautiful girls.
DLSM	These!
MATTES	these faithful wounds that kiss good ladies's chests, being black, which puts us in my mind.

Outros exemplos de textos transferidos por MATTES, DLSM e Style Transformer, na tarefa de transferência de sentimento, foram inseridos no Apêndice A.2.

Relembrando a questão 3 de pesquisa enunciada no capítulo 1, tem-se: *Como um modelo de TST que se vale de um modelo de linguagem mascarado se compara a outros modelos no estado da arte?* Buscando respondê-la, esse capítulo descreveu uma abordagem experimental para avaliar o modelo proposto, assim como para compará-lo com outros modelos do estado da arte. Seguindo esses procedimentos experimentais, esse capítulo mostrou os resultados quantitativos obtidos, os resultados das avaliações com pessoas e as comparações do método proposto com outros trabalhos relacionados na tarefa de TST não-supervisionada.

Capítulo 5

Conclusões

Essa dissertação propôs um novo método baseado em aprendizado de máquina para executar a tarefa de transferência estilo textual não supervisionada, onde se tem disponível para treinamento somente a sentença e seu respectivo estilo textual. Interpretar um texto e convertê-lo para o estilo desejado é uma habilidade fundamental para comunicação entre pessoas e dotar máquinas com essas habilidades é necessário para inseri-las como partes nesse complexo processo que é a comunicação e a linguagem.

Nesta dissertação, discutiu-se como as tarefas de geração de textos em língua natural são difíceis de serem realizadas, devido à dificuldade se obter dados paralelos para treinamento e a própria dificuldade para avaliar a geração automática de textos. Propôs-se um método de treinamento que, até onde é sabido, é o único que usa as representações bidirecionais produzidas por um modelo de linguagem mascarado como rótulos para serem usados durante o treinamento da rede geradora, na tarefa de transferência de estilo textual.

A arquitetura geral do MATTES foi inspirada em um trabalho anterior, o *Style Transformer*. Apesar disso, os modos de treinamento são diferentes. MATTES adota como rótulo a distribuição de probabilidade fornecida por um modelo de linguagem mascarado para cada token da sentença de entrada, em uma parte de sua função de custo. Após a realização de diversos experimentos, os resultados indicaram que a aplicação dessa técnica de treinamento é benéfica para o desempenho do modelo, os quais obteve métricas de controle de estilo superiores aos outros do estado da arte e manteve a preservação do conteúdo tão alta quanto, nos dois tipos de avaliações realizadas (automáticas e humanas).

5.1 Limitações

A ausência de dados para treinamento ou a sua baixa quantidade, e o amplo espaço amostral a ser vasculhado no processo do otimização da função de custo são condições de contorno que formam uma barreira que inviabilizam a melhora do desempenho do treinamento, após um certo ponto.

No que diz respeito aos resultados obtidos, um ponto de alerta é o fato de não se ter adotado uma métrica única de avaliação global dos textos gerados, ocasionando uma certa miopia avaliativa. Avaliaram-se individualmente três dimensões distintas dos textos gerados, e não se adotou uma métrica única de avaliação. Além disso, para chancelar nossa hipótese de que uso de representações bidirecionais ao longo do treinamento melhora a qualidade da geração de texto na tarefa de TST não supervisionada, experimentos mais vastos usando a técnica de extração de conhecimento, em diversas tarefas de transferência de estilo, poderiam ser realizados.

5.2 Trabalhos Futuros

Como próximos passos, o desempenho de nossa modelagem será testada em outros conjuntos de dados e tarefas, simplificação textual por exemplo, para chancelar a capacidade de generalização do método proposto. Além disso, buscar-se-á inserir a técnica de extração de conhecimento em outros modelos que adotam o componente de *back-translation* na função de custo para confirmar que essa técnica melhora a geração do textos na tarefa não-supervisionada de transferência de estilo textual. A adoção da técnica de destilação de conhecimento também pode ser testada no componente de reconstrução de sentença da rede geradora proposta. Considerando a flexibilidade da arquitetura, no futuro, planeja-se adaptar o modelo proposto para uma configuração que permita fazer a transferência para diversos domínios estilísticos e também que permita controlar múltiplos atributos, adentrando em áreas ainda pouco estudadas. Outra ideia promissora, que vem de estudos em tradução automática, é otimizar diretamente uma métrica de preservação de conteúdo durante o treinamento do modelo. Por fim, vislumbra-se treinar um modelo usando MATTES em conjuntos de dados na língua portuguesa, em tarefas como imitação autoral, geração de paráfrases e adaptação temporal (atualização da ortografia) de textos de medicina do século XVIII.

Por fim, reforçam-se os ganhos oriundos das tarefas de geração de textos em língua na-

tural. Com o aumento da presença de máquinas como partes do processo de comunicação, espera-se que essa sub-área do Processamento de Língua Natural continue ganhando destaque no mundo acadêmico e industrial. O avanço dos algoritmos de aprendizado de máquina que atuam ou são compatíveis com a tarefa de transferência de estilo textual está possibilitando a criação de modelos capazes de controlar, de forma suave, os atributos presentes nos textos. Acredita-se que, muito em breve, os textos gerados por máquinas ficarão indistinguíveis dos textos gerados por pessoas.

Referências

[1] Ahmad, Z.; S., M. N.; Ekbal, A.; Bhattacharyya, P. Tweet to news conversion: An investigation into unsupervised controllable text generation. *CoRR abs/2008.09333* (2020).

[2] Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings* (2015), Y. Bengio and Y. LeCun, Eds.

[3] Bender, E. M.; Koller, A. Climbing towards NLU: on meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020* (2020), D. Jurafsky, J. Chai, N. Schluter, and J. R. Tetreault, Eds., Association for Computational Linguistics, pp. 5185–5198.

[4] Bengio, Y.; Ducharme, R.; Vincent, P.; Janvin, C. A neural probabilistic language model. *The journal of machine learning research 3* (2003), 1137–1155.

[5] Bird, S. NLTK: the natural language toolkit. In *ACL 2006, 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, Sydney, Australia, 17-21 July 2006* (2006), N. Calzolari, C. Cardie, and P. Isabelle, Eds., The Association for Computer Linguistics.

[6] Boser, B. E.; Guyon, I. M.; Vapnik, V. N. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory* (1992), pp. 144–152.

[7] Brown, P. F.; Della Pietra, S. A.; Della Pietra, V. J.; Mercer, R. L. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics 19*, 2 (1993), 263–311.

[8] Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; Amodei, D. Language models are few-shot learners. In *Advances in Neural Information Processing Systems* (2020), H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33, Curran Associates, Inc., pp. 1877–1901.

[9] Can, F.; Patton, J. M. Change of writing style with time. *Computers and the Humanities 38*, 1 (2004), 61–82.

[10] Chen, L.; Dai, S.; Tao, C.; Shen, D.; Gan, Z.; Zhang, H.; Zhang, Y.; Zhang, R.; Wang, G.; Carin, L. Adversarial text generation via feature-mover's distance. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems* (Red Hook, NY, USA, 2018), NIPS'18, Curran Associates Inc., p. 4671–4682.

[11] Chen, X.; Zhang, S.; Shen, G.; Deng, Z.-H.; Yun, U. Towards unsupervised text multi-style transfer with parameter-sharing scheme. *Neurocomputing 426* (2021), 227–234.

[12] Chen, Y.; Gan, Z.; Cheng, Y.; Liu, J.; Liu, J. Distilling knowledge learned in BERT for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020* (2020), D. Jurafsky, J. Chai, N. Schluter, and J. R. Tetreault, Eds., Association for Computational Linguistics, pp. 7893–7905.

[13] Cunha, C.; Cintra, L. *Nova gramática do português contemporâneo*. LEXIKON Editora Digital ltda, 1985.

[14] Cybenko, G. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems 2*, 4 (1989), 303–314.

[15] Dai, N.; Liang, J.; Qiu, X.; Huang, X. Style transformer: Unpaired text style transfer without disentangled latent representation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers* (2019), A. Korhonen, D. R. Traum, and L. Màrquez, Eds., Association for Computational Linguistics, pp. 5997–6007.

[16] Dantas, S. Pay attention - explicando o mecanismo de atenção. `https://lamfo-unb.github.io/2019/05/01/Pay-attention-Explicando-o-mecanismo-de-Atencao/`. Accessed: 2021-08-28.

[17] Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (2019), pp. 4171–4186.

[18] Dolan, W. B.; Brockett, C. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing, IWP@IJCNLP 2005, Jeju Island, Korea, October 2005, 2005* (2005), Asian Federation of Natural Language Processing.

[19] dos Santos, C. N.; Melnyk, I.; Padhi, I. Fighting offensive language on social media with unsupervised text style transfer. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers* (2018), I. Gurevych and Y. Miyao, Eds., Association for Computational Linguistics, pp. 189–194.

[20] Edunov, S.; Baevski, A.; Auli, M. Pre-trained language model representations for language generation. In *Proceedings of NAACL-HLT* (2019), pp. 4052–4059.

[21] Fan, A.; Lewis, M.; Dauphin, Y. N. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers* (2018), I. Gurevych and Y. Miyao, Eds., Association for Computational Linguistics, pp. 889–898.

[22] Fu, Z.; Tan, X.; Peng, N.; Zhao, D.; Yan, R. Style transfer in text: Exploration and evaluation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18)* (2018), AAAI Press, pp. 663–670.

[23] Gatt, A.; Krahmer, E. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *J. Artif. Intell. Res. 61* (2018), 65–170.

[24] Gong, H.; Bhat, S.; Wu, L.; Xiong, J.; Hwu, W.-m. Reinforcement learning based text style transfer without parallel training corpus. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (Minneapolis, Minnesota, June 2019), Association for Computational Linguistics, pp. 3168–3180.

[25] Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[26] Goodfellow, I.; 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 27* (2014).

[27] Goyal, N.; Srinivasan, B. V.; N, A.; Sancheti, A. Multi-style transfer with discriminative feedback on disjoint corpus. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (Online, June 2021), Association for Computational Linguistics, pp. 3500–3510.

[28] Grishman, R. *Computational linguistics: an introduction*. Cambridge University Press, 1986.

[29] Gu, J.; Bradbury, J.; Xiong, C.; Li, V. O.; Socher, R. Non-autoregressive neural machine translation. In *International Conference on Learning Representations* (2018).

[30] Gururangan, S.; Marasovic, A.; Swayamdipta, S.; Lo, K.; Beltagy, I.; Downey, D.; Smith, N. A. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020* (2020), Association for Computational Linguistics, pp. 8342–8360.

[31] He, J.; Wang, X.; Neubig, G.; Berg-Kirkpatrick, T. A probabilistic formulation of unsupervised text style transfer. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020* (2020), OpenReview.net.

[32] Hendrycks, D.; Gimpel, K. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415* (2016).

[33] Hinton, G.; Vinyals, O.; Dean, J. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* (2015).

[34] Hinton, G. E.; Salakhutdinov, R. R. Reducing the dimensionality of data with neural networks. *science 313*, 5786 (2006), 504–507.

[35] Hinton, G. E.; Zemel, R. S. Autoencoders, minimum description length and helmholtz free energy. In *Proceedings of the 6th International Conference on Neural Information Processing Systems* (San Francisco, CA, USA, 1993), NIPS'93, Morgan Kaufmann Publishers Inc., p. 3–10.

[36] Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural computation 9*, 8 (1997), 1735–1780.

[37] Hornik, K.; Stinchcombe, M.; White, H. Multilayer feedforward networks are universal approximators. *Neural networks 2*, 5 (1989), 359–366.

[38] Hovy, E. Generating natural language under pragmatic constraints. *Journal of Pragmatics 11*, 6 (1987), 689–719.

[39] Howard, J.; Ruder, S. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers* (2018), I. Gurevych and Y. Miyao, Eds., Association for Computational Linguistics, pp. 328–339.

[40] Hu, Z.; Lee, R. K.; Aggarwal, C. C. Text style transfer: A review and experiment evaluation. *CoRR abs/2010.12742* (2020).

[41] Hu, Z.; Yang, Z.; Liang, X.; Salakhutdinov, R.; Xing, E. P. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017* (2017), D. Precup and Y. W. Teh, Eds., vol. 70 of *Proceedings of Machine Learning Research*, PMLR, pp. 1587–1596.

[42] Jain, P.; Mishra, A.; Azad, A. P.; Sankaranarayanan, K. Unsupervised controllable text formalization. *Proceedings of the AAAI Conference on Artificial Intelligence 33*, 01 (Jul. 2019), 6554–6561.

[43] Jang, E.; Gu, S.; Poole, B. Categorical reparameterization with gumbel-softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings* (2017), OpenReview.net.

[44] John, V.; Mou, L.; Bahuleyan, H.; Vechtomova, O. Disentangled representation learning for non-parallel text style transfer. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Volume 1: Long Papers* (2019), A. Korhonen, D. R. Traum, and L. Màrquez, Eds., Association for Computational Linguistics, pp. 424–434.

[45] Jurafsky, D.; Martin, J. H. *Speech and language processing - an introduction to natural language processing, computational linguistics, and speech recognition.* Prentice Hall series in artificial intelligence, 2000.

[46] Kim, H.; Sohn, K.-A. How positive are you: Text style transfer using adaptive style embedding. In *Proceedings of the 28th International Conference on Computational Linguistics* (Barcelona, Spain (Online), Dec. 2020), International Committee on Computational Linguistics, pp. 2115–2125.

[47] Kim, S.; Lee, J.; Gweon, G. Comparing data from chatbot and web surveys: Effects of platform and conversational style on survey response quality. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, CHI 2019, Glasgow, Scotland, UK, May 04-09, 2019* (2019), S. A. Brewster, G. Fitzpatrick, A. L. Cox, and V. Kostakos, Eds., ACM, p. 86.

[48] Kim, Y. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL* (2014), A. Moschitti, B. Pang, and W. Daelemans, Eds., ACL, pp. 1746–1751.

[49] Kingma, D. P.; Welling, M. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings* (2014), Y. Bengio and Y. LeCun, Eds.

[50] Krishna, K.; Wieting, J.; Iyyer, M. Reformulating unsupervised style transfer as paraphrase generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020* (2020), B. Webber, T. Cohn, Y. He, and Y. Liu, Eds., Association for Computational Linguistics, pp. 737–762.

[51] 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 and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (Hong Kong, China, Nov. 2019), Association for Computational Linguistics, pp. 3579–3584.

[52] Lai, G.; Xie, Q.; Liu, H.; Yang, Y.; Hovy, E. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (Copenhagen, Denmark, Sept. 2017), Association for Computational Linguistics, pp. 785–794.

[53] Lample, G.; Subramanian, S.; Smith, E. M.; Denoyer, L.; Ranzato, M.; Boureau, Y. Multiple-attribute text rewriting. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019* (2019), OpenReview.net.

[54] Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; Soricut, R. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020* (2020), OpenReview.net.

[55] LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *nature 521*, 7553 (2015), 436–444.

[56] Li, J.; Jia, R.; He, H.; Liang, P. Delete, retrieve, generate: a simple approach to sentiment and style transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, Volume 1 (Long Papers)* (2018), Association for Computational Linguistics, pp. 1865–1874.

[57] Liu, D.; Fu, J.; Zhang, Y.; Pal, C.; Lv, J. Revision in continuous space: Unsupervised text style transfer without adversarial learning. *Proceedings of the AAAI Conference on Artificial Intelligence 34*, 05 (Apr. 2020), 8376–8383.

[58] Liu, Y.; Neubig, G.; Wieting, J. On learning text style transfer with direct rewards. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (Online, June 2021), Association for Computational Linguistics, pp. 4262–4273.

[59] Liu, Z.; Lin, Y.; Sun, M. *Representation learning for natural language processing.* Springer Nature, 2020.

[60] Logeswaran, L.; Lee, H.; Bengio, S. Content preserving text generation with attribute controls. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems* (Red Hook, NY, USA, 2018), NIPS'18, Curran Associates Inc., p. 5108–5118.

[61] Luo, F.; Li, P.; Zhou, J.; Yang, P.; Chang, B.; Sun, X.; Sui, Z. A dual reinforcement learning framework for unsupervised text style transfer. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19* (7 2019), International Joint Conferences on Artificial Intelligence Organization, pp. 5116–5122.

[62] Luong, T.; Pham, H.; Manning, C. D. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015* (2015), L. Màrquez, C. Callison-Burch, J. Su, D. Pighin, and Y. Marton, Eds., The Association for Computational Linguistics, pp. 1412–1421.

[63] Malmi, E.; Severyn, A.; Rothe, S. Unsupervised text style transfer with padded masked language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November*

16-20, 2020 (2020), B. Webber, T. Cohn, Y. He, and Y. Liu, Eds., Association for Computational Linguistics, pp. 8671–8680.

[64] Manning, C.; Schutze, H. *Foundations of statistical natural language processing*. MIT press, 1999.

[65] Marouzeau, J. *Précis de stylistique française*. Masson, 1963.

[66] McDonald, D. D.; Pustejovsky, J. A computational theory of prose style for natural language generation. In *Second Conference of the European Chapter of the Association for Computational Linguistics* (1985).

[67] Mishra, A.; Tater, T.; Sankaranarayanan, K. A modular architecture for unsupervised sarcasm generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing EMNLP-IJCNLP* (Hong Kong, China, Nov. 2019), Association for Computational Linguistics, pp. 6144–6154.

[68] Mitchell, T. M. *Machine Learning*. McGraw-Hill, New York, 1997.

[69] Mueller, J.; Gifford, D.; Jaakkola, T. Sequence to better sequence: Continuous revision of combinatorial structures. In *Proceedings of the 34th International Conference on Machine Learning* (06–11 Aug 2017), D. Precup and Y. W. Teh, Eds., vol. 70 of *Proceedings of Machine Learning Research*, PMLR, pp. 2536–2544.

[70] Nadkarni, P. M.; Ohno-Machado, L.; Chapman, W. W. Natural language processing: an introduction. *Journal of the American Medical Informatics Association 18*, 5 (09 2011), 544–551.

[71] Nilsson, N. J. *Principles of artificial intelligence*. Morgan Kaufmann, 2014.

[72] Papineni, K.; Roukos, S.; Ward, T.; Zhu, W. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA* (2002), ACL, pp. 311–318.

[73] Park, S.; Hwang, S.-w.; Chen, F.; Choo, J.; Ha, J.-W.; Kim, S.; Yim, J. Paraphrase diversification using counterfactual debiasing. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2019), vol. 33:01, pp. 6883–6891.

[74] Parra, G., et al. Automated writing evaluation tools in the improvement of the writing skill. *International Journal of Instruction 12*, 2 (2019), 209–226.

[75] Peters, M.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (2018), pp. 2227–2237.

[76] Peters, M. E.; Ruder, S.; Smith, N. A. To tune or not to tune? adapting pretrained representations to diverse tasks. In *Proceedings of the 4th Workshop on Representation Learning for NLP, RepL4NLP@ACL 2019, Florence, Italy, August*

2, 2019 (2019), I. Augenstein, S. Gella, S. Ruder, K. Kann, B. Can, J. Welbl, A. Conneau, X. Ren, and M. Rei, Eds., Association for Computational Linguistics, pp. 7–14.

[77] Phuong, M.; Lampert, C. Towards understanding knowledge distillation. In *International Conference on Machine Learning* (2019), PMLR, pp. 5142–5151.

[78] Prabhumoye, S.; Tsvetkov, Y.; Salakhutdinov, R.; Black, A. W. Style transfer through back-translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers* (2018), I. Gurevych and Y. Miyao, Eds., Association for Computational Linguistics, pp. 866–876.

[79] Radford, A.; Metz, L.; Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings* (2016), Y. Bengio and Y. LeCun, Eds.

[80] Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf* (2018).

[81] Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language models are unsupervised multitask learners, 2019.

[82] Rajpurkar, P.; Zhang, J.; Lopyrev, K.; Liang, P. Squad: 100, 000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016* (2016), J. Su, X. Carreras, and K. Duh, Eds., The Association for Computational Linguistics, pp. 2383–2392.

[83] Rogers, A.; Kovaleva, O.; Rumshisky, A. A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics 8* (2020), 842–866.

[84] Ruder, S.; Peters, M. E.; Swayamdipta, S.; Wolf, T. Transfer learning in natural language processing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials* (2019), pp. 15–18.

[85] Rush, A. M.; Chopra, S.; Weston, J. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015* (2015), L. Màrquez, C. Callison-Burch, J. Su, D. Pighin, and Y. Marton, Eds., The Association for Computational Linguistics, pp. 379–389.

[86] Saini, N.; Trivedi, D.; Khare, S.; Dhamecha, T. I.; Jyothi, P.; Bharadwaj, S.; Bhattacharyya, P. Disfluency correction using unsupervised and semi-supervised learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL*

2021, Online, April 19 - 23, 2021 (2021), P. Merlo, J. Tiedemann, and R. Tsarfaty, Eds., Association for Computational Linguistics, pp. 3421–3427.

[87] Sennrich, R.; Haddow, B.; Birch, A. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers* (2016), The Association for Computer Linguistics.

[88] Shen, T.; Lei, T.; Barzilay, R.; Jaakkola, T. S. Style transfer from non-parallel text by cross-alignment. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA* (2017), I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, Eds., pp. 6830–6841.

[89] Slama-Cazacu, T. Langage et contexte. *Les Etudes Philosophiques 19*, 1 (1964).

[90] Sudhakar, A.; Upadhyay, B.; Maheswaran, A. "transforming"delete, retrieve, generate approach for controlled text style transfer. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019* (2019), Association for Computational Linguistics, pp. 3267–3277.

[91] Sun, S.; Cheng, Y.; Gan, Z.; Liu, J. Patient knowledge distillation for BERT model compression. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (Hong Kong, China, Nov. 2019), Association for Computational Linguistics, pp. 4323–4332.

[92] Surya, S.; Mishra, A.; Laha, A.; Jain, P.; Sankaranarayanan, K. Unsupervised neural text simplification. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers* (2019), A. Korhonen, D. R. Traum, and L. Màrquez, Eds., Association for Computational Linguistics, pp. 2058–2068.

[93] Sutskever, I.; Vinyals, O.; Le, Q. V. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada* (2014), Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., pp. 3104–3112.

[94] Tian, Y.; Hu, Z.; Yu, Z. Structured content preservation for unsupervised text style transfer. *arXiv preprint arXiv:1810.06526* (2018).

[95] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA* (2017), I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, Eds., pp. 5998–6008.

[96] Wang, K.; Hua, H.; Wan, X. Controllable unsupervised text attribute transfer via editing entangled latent representation. *Advances in Neural Information Processing Systems 32* (2019), 11036–11046.

[97] Wang, Y.; Liao, Y.; Wu, Y.; Chang, L. Conditional random field-based parser and language model for tradi-tional chinese spelling checker. In *Proceedings of the Seventh SIGHAN Workshop on Chinese Language Processing, SIGHAN@IJCNLP 2013, Nagoya, Japan, October 14-18, 2013* (2013), L. Yu, Y. Tseng, J. Zhu, and F. Ren, Eds., Asian Federation of Natural Language Processing, pp. 69–73.

[98] Wen, T.; Gasic, M.; Mrksic, N.; Su, P.; Vandyke, D.; Young, S. J. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015* (2015), L. Màrquez, C. Callison-Burch, J. Su, D. Pighin, and Y. Marton, Eds., The Association for Computational Linguistics, pp. 1711–1721.

[99] Wieting, J.; Berg-Kirkpatrick, T.; Gimpel, K.; Neubig, G. Beyond BLEU: training neural machine translation with semantic similarity. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers* (2019), A. Korhonen, D. R. Traum, and L. Màrquez, Eds., Association for Computational Linguistics, pp. 4344–4355.

[100] Wieting, J.; Gimpel, K. ParaNMT-50M: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (Melbourne, Australia, July 2018), Association for Computational Linguistics, pp. 451–462.

[101] Williams, A.; Nangia, N.; Bowman, S. R. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, Volume 1 (Long Papers)* (2018), M. A. Walker, H. Ji, and A. Stent, Eds., Association for Computational Linguistics, pp. 1112–1122.

[102] Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning 8*, 3-4 (1992), 229–256.

[103] Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; Davison, J.; Shleifer, S.; von Platen, P.; Ma, C.; Jernite, Y.; Plu, J.; Xu, C.; Scao, T. L.; Gugger, S.; Drame, M.; Lhoest, Q.; Rush, A. M. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020* (2020), Q. Liu and D. Schlangen, Eds., Association for Computational Linguistics, pp. 38–45.

[104] Wu, C.; Ren, X.; Luo, F.; Sun, X. A hierarchical reinforced sequence operation method for unsupervised text style transfer. In *Proceedings of the 57th Conference*

of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers (2019), A. Korhonen, D. R. Traum, and L. Màrquez, Eds., Association for Computational Linguistics, pp. 4873–4883.

[105] Wu, X.; Zhang, T.; Zang, L.; Han, J.; Hu, S. Mask and infill: Applying masked language model for sentiment transfer. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019* (2019), S. Kraus, Ed., ijcai.org, pp. 5271–5277.

[106] Xu, J.; Sun, X.; Zeng, Q.; Zhang, X.; Ren, X.; Wang, H.; Li, W. Unpaired sentiment-to-sentiment translation: A cycled reinforcement learning approach. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Volume 1: Long Papers* (2018), I. Gurevych and Y. Miyao, Eds., Association for Computational Linguistics, pp. 979–988.

[107] Xu, P.; Cheung, J. C. K.; Cao, Y. On variational learning of controllable representations for text without supervision. In *International Conference on Machine Learning* (2020), PMLR, pp. 10534–10543.

[108] Xu, W.; Ritter, A.; Dolan, B.; Grishman, R.; Cherry, C. Paraphrasing for style. In *COLING 2012, 24th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, 8-15 December 2012, Mumbai, India* (2012), M. Kay and C. Boitet, Eds., Indian Institute of Technology Bombay, pp. 2899–2914.

[109] Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J. G.; Salakhutdinov, R.; Le, Q. V. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada* (2019), H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, Eds., pp. 5754–5764.

[110] Yang, Z.; Hu, Z.; Dyer, C.; Xing, E. P.; Berg-Kirkpatrick, T. Unsupervised text style transfer using language models as discriminators. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada* (2018), S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., pp. 7298–7309.

[111] Yi, X.; Liu, Z.; Li, W.; Sun, M. Text style transfer via learning style instance supported latent space. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20* (7 2020), C. Bessiere, Ed., International Joint Conferences on Artificial Intelligence Organization, pp. 3801–3807. Main track.

[112] Yin, D.; Huang, S.; Dai, X.-Y.; Chen, J. Utilizing non-parallel text for style transfer by making partial comparisons. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19* (7 2019), International Joint Conferences on Artificial Intelligence Organization, pp. 5379–5386.

[113] Zhang, T.; Kishore, V.; Wu, F.; Weinberger, K. Q.; Artzi, Y. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020* (2020), OpenReview.net.

[114] Zhang, Y.; Ding, N.; Soricut, R. SHAPED: Shared-private encoder-decoder for text style adaptation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (New Orleans, Louisiana, June 2018), Association for Computational Linguistics, pp. 1528–1538.

[115] Zhang, Y.; Xu, J.; Yang, P.; Sun, X. Learning sentiment memories for sentiment modification without parallel data. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018* (2018), E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, Eds., Association for Computational Linguistics, pp. 1103–1108.

[116] Zhang, Z.; Ren, S.; Liu, S.; Wang, J.; Chen, P.; Li, M.; Zhou, M.; Chen, E. Style transfer as unsupervised machine translation. *arXiv preprint arXiv:1808.07894* (2018).

[117] Zhao, J.; Kim, Y.; Zhang, K.; Rush, A.; LeCun, Y. Adversarially regularized autoencoders. In *Proceedings of the 35th International Conference on Machine Learning* (10–15 Jul 2018), J. Dy and A. Krause, Eds., vol. 80 of *Proceedings of Machine Learning Research*, PMLR, pp. 5902–5911.

[118] Zhao, Y.; Bi, W.; Cai, D.; Liu, X.; Tu, K.; Shi, S. Language style transfer from sentences with arbitrary unknown styles. *arXiv preprint arXiv:1808.04071* (2018).

[119] Zhou, C.; Chen, L.; Liu, J.; Xiao, X.; Su, J.; Guo, S.; Wu, H. Exploring contextual word-level style relevance for unsupervised style transfer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (Online, July 2020), Association for Computational Linguistics, pp. 7135–7144.

APÊNDICE A

A.1 Escolha de Hiperparâmetros

Com intuito de tornar os experimentos conduzidos nessa dissertação reproduzíveis, lista-se os hiperparâmetros usados no treinamento dos modelos que alcançaram melhor desempenho, mostrados na Tabela 4.1, em ambas as tarefas.

Tarefa de Transferência de Sentimento

- $T_{KD} = 10$
- $\alpha = 0.5$
- $(n_d, n_f) = (7, 5)$
- Função de custo do Transformer Mascarado:

$$0.15\mathcal{L}_{self}(\theta) + 0.3\mathcal{L}_{KD}(\theta) + \mathcal{L}_{adv}(\rho) \tag{A.1}$$

- Dropout de palavras = 0.2
- Tamanho do lote = 64 sentenças de cada estilo
- Máximo comprimento da sentença = 32
- Taxa de aprendizado da rede geradora = 0.0001
- Taxa de aprendizado da rede discriminadora = 0.0001

Tarefa de Imitação Autoral

- $T_{KD} = 5$

- $\alpha = 0.5$
- $(n_d,n_f) = (7, 5)$
- Função de custo do Transformer Mascarado:

$$0.2\mathcal{L}_{self}(\theta) + 0.5\mathcal{L}_{KD}(\theta) + \mathcal{L}_{adv}(\rho) \tag{A.2}$$

- Dropout de palavras = 0.3
- Tamanho do lote = 32 sentenças de cada domínio
- Máximo comprimento da sentença = 64
- Taxa de aprendizado da rede geradora = 0.0001
- Taxa de aprendizado da rede discriminadora = 0.0001

A.2 Exemplos de Transferência de Sentimento

Exemplos de sentenças transferidas na tarefa de transferência de sentimento comparando MATTES, DLSM e *Style Transformer* estão na Tabela A.1.

Modelo	**Negative to Positive**
Source	i guess she wasn't happy that we were asking the prices.
Deep Lat	i love it and i love the happy hour we were expecting the prices.
Style Tranf.	i recommend she was always happy that we were asking the prices.
MATTES	i hope she's happy that we were asking the prices.
Source	went to the sunday brunch to celebrate our daughter's college graduation.
Deep Lat	went to the sunday brunch to celebrate our daughter's college favorite.
Style Tranf.	easy to the sunday brunch to celebrate our daughter's college graduation.
MATTES	i went to the sunday brunch to celebrate our daughter's college graduation.
Source	you can not judge people based on appearance.
Deep Lat	you can definitely count based on time on appearance.
Style Tranf.	you can definitely judge people based on appearance.
MATTES	you can definitely judge people based on appearance.
Source	did they not have a fountain machine on site?
Deep Lat	lots of fun, outdoor seating on the site!
Style Tranf.	did they well have a fountain machine on site!
MATTES	always they have a fountain bands on site!
	Positive to Negative
Source	rick is a seriously cool guy!
Deep Lat	ugh.
Style Tranf.	rick was a seriously over guy?
MATTES	rick is a seriously sad guy!
Source	i highly recommend e & m painting.
Deep Lat	i wouldn't recommend & m m.
Style Tranf.	i won't be e & m painting.
MATTES	i would not e & m painting.
Source	we recommend imports & american auto service to everyone we know.
Deep Lat	we would rather buy an american auto service to everyone we know.
Style Tranf.	we wouldn't billed & american auto service to everyone we know.
MATTES	we disliked treatment & american auto service to me we know.
Source	loved the burgers, i had the jalapeo ranch burger it was really tasty.
Deep Lat	however, the burgers i had the jalapeo ranch burger it was really disappointing.
Style Tranf.	ordered the burgers, i had the jalapeo ranch burger it was really tasty.
MATTES	she said the burgers, i had the jalapeo ranch burger was really bland.

Tabela A.1: Sentenças transferidas na Tarefa de Transferência de Sentimento

A.3 Formulário Gerado para Avaliação por Pessoas

Inteligência Artificial - UFF

*Obrigatório

Experimento de transferência de sentimento

Como resultado de pesquisa acadêmica na UFF, foi desenvolvido um método de tranferência de sentimento, que converte o sentimento de sentenças na língua INGLESA do NEGATIVO para o POSITIVO e do POSITIVO para o NEGATIVO. A ideia é o modelo converter, por exemplo, uma sentença positiva como "This restaurant is fantastic" para "This restaurant is terrible". Com o intuito de comparar o método desenvolvido com outros dois métodos da literatura , gostaríamos de solicitar a sua avaliação em algumas sentenças produzidas pelos três modelos.

Cada alternativa corresponde a um método. Se a resposta que você achar correta se REPETIR, marque qualquer uma delas, pois sua pontuação será atribuída a todos os métodos com a mesma resposta. Caso as três opções sejam igualmente boas ou igualmente ruins, por favor, selecione a opção NO PREFERENCE.

Na tarefa de transferência de sentimento, almeja-se que os textos gerados tenham o sentimento desejado, preservem o conteúdo semântico da mensagem original e sejam fluentes. Serão feitas perguntas para avaliar essas três dimensões.

O formulário possui, para avaliação, apenas 2 conversões do NEGATIVO para o POSITIVO, e 2 conversões do POSITIVO para o NEGATIVO

Serão necessários entre 5 e 10 minutos para o preenchimento. Desde já, agradecemos muito!

Essa atividade faz parte da pesquisa do aluno Arthur Scalercio, mestrando do curso de Ciência da Computação na Universidade Federal Fluminense (IC/UFF), orientado pela professora Aline Paes (IC/UFF). As informações coletadas serão destinadas exclusivamente à validação dos modelos obtidos e o anonimato dos participantes será preservado. A atividade pode ser interrompida a qualquer momento, segundo a sua disponibilidade e vontade.

Para mais informações entrar em contato através dos emails:
- arthurscalercio@gmail.com
- alinepaes@ic.uff.br

E-mail:

Sua resposta

Qual o seu conhecimento sobre a área de Processamento de Linguagem Natural? *

- ○ Nenhum
- ○ Entendo pouco
- ○ Entendo razoavelmente
- ◉ Avançado
- ○ Especialista

Escolha um número entre 1 e 15 para iniciar *

11

Próxima

Nunca envie senhas pelo Formulários Google.

Inteligência Artificial - UFF

*Obrigatório

Conversão de sentimento NEGATIVO para POSITIVO e POSITIVO para NEGATIVO

Pede-se que seja avaliada qual dos 3 modelos está performando melhor na tarefa de transferir o sentimento de uma sentença. Nessa tarefa, deseja-se que o modelo mantenha o conteúdo da sentença de entrada e altere somente o sentimento da sentença. Caso as três opções sejam igualmente boas ou igualmente ruins, por favor, selecione a opção NO PREFERENCE. Solicita-se que essa avaliação ocorra conforme as perguntas a seguir:

Which sentence has the most positive sentiment toward the following sentence: "an old dude did my pedicure." *

- ◯ No Preference
- ◯ an authentic dude did my pedicure.
- ◯ an fresh dude did my pedicure.
- ◯ an old dude did my pedicure.

Which sentence retains most content from the sentence: "an old dude did my pedicure." *

- ◯ an fresh dude did my pedicure.
- ◯ an old dude did my pedicure.
- ◯ No Preference
- ◯ an authentic dude did my pedicure.

Which sentence is the most fluent one? *

- an fresh dude did my pedicure.
- an old dude did my pedicure.
- an authentic dude did my pedicure.
- No Preference

Which sentence has the most positive sentiment toward the following sentence: "there was only meat and bread." *

- there was best meat and bread.
- amazing service.
- No Preference
- there was also meat and bread.

Which sentence retains most content from the sentence: "there was only meat and bread." *

- there was also meat and bread.
- there was best meat and bread.
- amazing service.
- No Preference

Which sentence is the most fluent one? *

- No Preference
- there was also meat and bread.
- amazing service.
- there was best meat and bread.

Which sentence has the most negative sentiment toward the following sentence: "its quiet and nice people are here." *

- No Preference
- its dingy and disappointing people are here.
- its ugly and poor people here are.
- its cold and rude people are here.

Which sentence retains most content from the sentence: "its quiet and nice people are here." *

- No Preference
- its ugly and poor people here are.
- its cold and rude people are here.
- its dingy and disappointing people are here.

Which sentence is the most fluent one? *

- its cold and rude people are here.
- its dingy and disappointing people are here.
- its ugly and poor people here are.
- No Preference

Which sentence has the most negative sentiment toward the following sentence: "blue corn tacos with chicken were excellent." *

- blue corn tacos with chicken were excellent.
- No Preference
- blue corn tacos with chicken were horrible.
- blue corn tacos with chicken were nothing special.

Which sentence retains most content from the sentence: "blue corn tacos with chicken were excellent." *

- No Preference
- blue corn tacos with chicken were excellent.
- blue corn tacos with chicken were nothing special.
- blue corn tacos with chicken were horrible.

Which sentence is the most fluent one? *

- ◯ blue corn tacos with chicken were horrible.
- ◯ No Preference
- ◯ blue corn tacos with chicken were excellent.
- ◯ blue corn tacos with chicken were nothing special.

Voltar Enviar