UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO

Uma Arquitetura Baseada em Aprendizado Profundo com Mecanismos de Atenção para a Tradução Contínua da Língua Brasileira de Sinais em Contextos sem Intérpretes

Diego Ramon Bezerra da Silva

Orientador: Prof. Dr. Luiz Marcos Garcia Gonçalves

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e de Computação da UFRN (área de concentração: Engenharia de Computação) como parte dos requisitos para obtenção do título de Doutor em Ciências.

Número de ordem PPgEEC: D380
Natal, RN, 20 de Dezembro de 2024

Universidade Federal do Rio Grande do Norte - UFRN
Sistema de Bibliotecas - SISBI
Catalogação de Publicação na Fonte. UFRN - Biblioteca Central Zila Mamede

Silva, Diego Ramon Bezerra da.
Uma arquitetura baseada em aprendizado profundo com Mecanismos de Atenção para a Tradução Contínua da Língua Brasileira de Sinais em Contextos sem Intérpretes / Diego Ramon Bezerra da Silva. - 2025.
99 f.: il.

Tese (doutorado) - Universidade Federal do Rio Grande do Norte, Centro de Tecnologia, Programa de Pós-Graduação em Engenharia Elétrica e de Computação, Natal, 2025.
Orientação: Prof. Dr. Luiz Marcos Garcia Gonçalves.

1. Acessibilidade - Tese. 2. Aprendizado profundo - Tese. 3. Libras - Tese. 4. Tradução automática - Tese. I. Gonçalves, Luiz Marcos Garcia. II. Título.

RN/UF/BCZM CDU 004.4

Elaborado por Jackeline dos Santos Pinheiro da Silva Maia Cavalcanti - CRB-15/317

Agradecimentos

Ao meu orientador Prof. Luiz Marcos Garcia Gonçalves e ao meu co-orientador (mesmo que informalmente) Prof. Tiago Maritan Ugolino de Araújo, sou grato pela orientação.

Aos colegas do projeto VLibras, em especial à equipe de surdos e intérpretes que auxiliaram no desenvolvimento da base de dados.

Aos demais colegas de pós-graduação, pelas críticas e sugestões.

À minha família pelo apoio durante esta jornada.

Resumo

No Brasil, os surdos representam cerca de 5% da população, aproximadamente 9.7 milhões de brasileiros. Apesar da Língua de Sinais Brasileira (Libras) ser uma das línguas oficiais do Brasil, o conhecimento e domínio de Libras entre pessoas não surdas é um obstáculo, gerando barreiras linguísticas no acesso a direitos básicos, especialmente no acesso a serviços de saúde. Isso motivou o desenvolvimento de políticas governamentais que obrigam os prestadores de serviços a fornecer intérpretes de Libras para viabilizar o acesso a esses serviços por parte da comunidade surda. Todavia, esse tipo de solução apresenta um custo de implantação e manutenção muito alto. Nessa perspectiva, se faz necessário o desenvolvimento de pesquisas e metodologias automatizadas para tradução automática de Libras. Assim, neste trabalho, propusemos uma metodologia para tradução contínua de Libras. A solução proposta não requer nenhum hardware adicional, baseando-se inteiramente em imagens ou sequências de imagens (vídeos). Além disso, foi introduzida um novo conjunto de dados para reconhecimento contínuo de Libras, contendo 10500 vídeos de 105 sentenças distintas no contexto de triagem clínica. Os experimentos computacionais obtiveram um WER de 21,62 e uma acurácia máxima de 92,68%, para um conjunto de testes com amostras e intérpretes nunca vistos pelo modelo durante o treinamento.

Palavras-chave: Acessibilidade, Libras, Aprendizado Profundo, Sequência-para-sequência, Tradução Automática.

Abstract

In Brazil, the deaf represent about 5% of the population approximately 9.7 million Brazilians. Despite the Brazilian Sign Language (Libras) being recognized as an official language in Brazil, the knowledge and mastery of Libras among the non-deaf is an obstacle, which ends up creating language barriers in accessing basic rights, especially in accessing health services. This has motivated the development of government policies that oblige service providers to provide Libras interpreters to enable access to these services by the deaf community. However, human-based approaches have a high implementation and maintenance cost. From this perspective, it is necessary to develop research and automated methodologies for automatic translation of Libras. Thus in this work, we proposed a methodology for continuous translation of Libras. The proposed methodology does not require any additional hardware, relying entirely on images or image sequences (videos). In addition, a new dataset for Continuous Sign Language Recognition (CSLR) was introduced, containing 10500 videos of 105 distinct sentences in the context of clinical screening. The evaluation experiments achieve a WER of 21.62, while maximum accuracy of about 92.68% for a set of tests with videos and signers never seen by the model during training.

Keywords: Accessibility, Libras, Deep Learning, Sequence-to-sequence, Machine Translation.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Símbolos e Abreviaturas

AM: Aprendizado de Máquina

ASL: Língua Americana de Sinais

BLEU: *Bilingual Evaluation Understudy*

CE: *Cross-Entropy*

CNN: *Convolutional Neural Network*

DL: *Deep Learning*

GPU: *Graphics Processing Unit*

GRU: *Gated Recurrent Neural Networks*

IA: Inteligência Artificial

IBGE: Instituto Brasileiro de Pesquisas Geográficas

Libras: Língua Brasileira de Sinais

LN: *Layer Normalization*

LS: Língua de Sinais

LSF: Língua Francesa de Sinais

LSTM: *Long-short Time Memory*

Mask: Máscara

MLP: *Multi Layer Perceptron*

MSA: *Multi-Head Self Attention*

MSE: *Mean Squared Error*

NLP: *Natural Language Processing*

PE: *Positional Encoding*

PLN: Processamento de Linguagem Natural

RNN: *Recurrent Neural Network*

RNT: Rede Neural Transformadora

ViT: *Vision Transformers*

ViViT: *Video Vision Transformers*

WER: *Word Error Rate*

WHO: *World Health Organization*

Capítulo 1

Introdução

A Língua de Sinais Brasileira (Libras) é o meio de comunicação principal e, muitas vezes, o único meio de comunicação disponível para muitas pessoas, resultando em barreiras de interação entre surdos e não-surdos. Essa assimetria gera problemas de acesso a serviços básicos que são, muitas vezes, direitos fundamentais do ser humano, como educação, segurança e principalmente saúde. A tradução automática de língua de sinais é um campo de estudo muito importante, pois viabiliza uma melhor integração e comunicação dessa parcela da população, proporcionando-lhes uma melhor inclusão social, garantindo, assim, oportunidades iguais de acesso a bens e serviços.

Neste contexto, o presente trabalho propõe uma solução baseada em vídeo para a tradução automática de Libras, com o objetivo de viabilizar uma comunicação bidirecional entre surdos e não-surdos e, com isso, melhorar o acesso de pessoas com deficiência auditiva a serviços básicos, particularmente no contexto de saúde.

Este capítulo apresenta a motivação deste trabalho na Seção 1.1, a definição do problema, as premissas e hipóteses de pesquisa na Seção 1.2, as contribuições na Seção 1.3, a definição do escopo do trabalho na Seção 1.4 e, por fim, na Seção 1.5 é apresentada a organização deste documento.

1.1 Motivação

Os surdos são considerados uma minoria linguística por utilizarem como principal meio de comunicação e expressão uma língua de sinais, o que gera barreiras no acesso à educação, saúde e lazer, além de limitar a interação social, motivando discussões e estudos com o objetivo de proporcionar inclusão social, educacional e de saúde a esta comunidade, uma vez que representam um quantitativo significativo da população (de Souza et al. 2017). No Mundo, de acordo com o Relatório Mundial sobre Audição (do inglês, *World Report on Hearing*) de 2021 da Organização Mundial da Saúde (do inglês, *World Health Organization* - WHO) existem aproximadamente 1,5 bilhões de pessoas com algum nível de deficiência auditiva, das quais 403 milhões apresentam perda auditiva significativa, o que representa cerca de 5,37% de toda a população mundial (WHO 2021)[1]. No Brasil, segundo o censo de 2010 do Instituto Brasileiro de Geografia e Estatística

[1] https://www.who.int/publications/i/item/world-report-on-hearing

(IBGE)[2], há aproximadamente 10 milhões de brasileiros com algum tipo de deficiência auditiva. Essa parcela representa cerca de 5,1% da população brasileira, dos quais 2,7 milhões apresentam um nível de surdez profunda.

Esses dados, aliados ao fato de que os surdos são uma comunidade minoritária linguística e cultural (de Souza et al. 2017) mostram que muitas pessoas surdas podem estar enfrentando barreiras de comunicação, acesso à informação e serviços públicos, especialmente serviços de saúde. Uma das razões para essa dificuldade é que as pessoas surdas comunicam-se naturalmente através de línguas gestuais-visuais (ver Figura 1.1), denominadas línguas de sinais, e as línguas orais representam apenas uma "segunda língua".

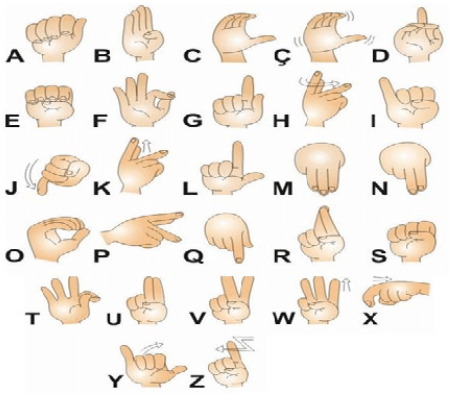

Figura 1.1: Alfabeto manual de Libras. Fonte: (Secco & Lino Ferreira da Silva Barros 2009)

As línguas de sinais (LS) são, de fato, línguas naturais e completas, com gramática, vocabulário e dialetos próprios, sendo dotadas de todas as funções sociais e mentais de uma língua falada, esse status foi consolidado desde a década de 60 com o trabalho pioneiro de (Stokoe 2005). As línguas de sinais, assim como todas as linguagens naturais, surgem naturalmente. Por exemplo, Libras não é uma representação mímica ou gestual da Língua Portuguesa, mas sim uma língua diferente, apresentando todos os elementos linguísticos necessários para tal classificação, delimitada por um contexto cultural próprio e diverso. No Brasil, formalmente, a Libras foi reconhecida como meio legal de comunicação e expressão em todo território nacional através da Lei Nº 10.436[3], de 24 de Abril de 2002.

As LS são naturais porque vêm da interação natural e do convívio entre surdos, sendo dotadas de capacidade para expressar e transmitir qualquer conceito concreto ou abstrato. Cada país apresenta sua própria língua de sinais. No Brasil, a Língua Brasileira de Sinais (Libras); nos Estados Unidos, a Língua Americana de Sinais (ASL); e na França, a Língua Francesa de Sinais (LSF), e assim por diante. Entretanto, as LS podem apresentar origens históricas e influências em outras LS, como, por exemplo, a Libras, que compartilha influências da LSF. Uma outra característica compartilhada com línguas orais é a existência

[2]https://censo2010.ibge.gov.br/

[3]http://www.planalto.gov.br/ccivil$_0$3/*leis*/2002/*l*10436.*htm*

de regionalismos. Isso significa que existem sinais que variam de acordo com a região, a idade e até o gênero de quem se comunica.

Problemas de comunicação interpessoal estão presentes em todo o sistema de saúde brasileiro e tornam-se mais significativos quando englobam barreiras culturais e de linguagem (Dias et al. 2017). Essa barreira é especialmente séria no contexto de saúde, já que uma comunicação eficiente entre o paciente e o profissional de saúde é uma etapa fundamental para o atendimento, diagnóstico e acompanhamento do tratamento médico. Apesar da Lei Nº 10.436, de 24 de Abril de 2002, tornar o ensino de Libras obrigatório para cursos de licenciatura e eletivo para os demais cursos das diversas áreas do conhecimento, são poucos os profissionais que estão aptos a manter uma comunicação de qualidade em Libras.

A problemática de integração da comunidade surda também ganhou bastante atenção do legislativo brasileiro. Projetos de leis como PLS 465/2017[4], ainda em tramitação no Congresso, altera a Lei N^o 10.048[5], de 8 de novembro de 2000, torna obrigatório a oferta de intérpretes de Libras em instituições públicas e concessionárias de serviços públicos de assistência à saúde. Além disso, a PLS 155/2017[6], obriga repartições públicas, empresas concessionárias de serviços públicos e instituições financeiras, os bancos, a contar com intérpretes da Língua Brasileira de Sinais.

Neste contexto, com a atuação do poder público, essa demanda social acarreta uma demanda por soluções prontas para o mercado, pois a não adequação às normas pode resultar em problemas jurídicos e impactos negativos para empresas que, porventura, não sigam as leis. Todavia, o cumprimento dessas leis está sujeito a problemas de implantação e manutenção das políticas de acessibilidade, tendo em vista que isso demandaria uma quantidade considerável de intérpretes humanos, além de um custo considerável para manter essa política de acessibilidade operacional. Portanto, é necessário um processo de pesquisa e desenvolvimento de novas ferramentas e tecnologias que possam suprir essa demanda de forma economicamente viável.

Portanto, a questão a ser investigada será a tradução automática contínua de Libras para a Língua Portuguesa, um tema bastante pesquisado na literatura brasileira e mundial, porém ainda em aberto, não havendo um método ou metodologia totalmente aceito e, principalmente, não existindo soluções que atendam o usuário final. Ou seja, a problemática investigada possui um forte apelo social, já que terá como objetivo final a facilitação do acesso a serviços de saúde por parte da população surda no Brasil.

O problema de tradução automática de Libras não é trivial além de toda a complexidade linguística. Pessoas diferentes têm velocidades diferentes de reprodução, aliado ao fato de que Libras é uma língua relativamente nova, sem uma norma culta estabelecida. Esses fatos adicionam dificuldades à solução da tradução contínua de Libras, pois exigem que um componente de inteligência seja altamente capaz de aprender as informações espaciais e temporais nas sequências de sinais, sendo um problema que apresenta intersecção entre as áreas de visão computacional e processamento de linguagem natural.

Assim, esta tese apresenta uma investigação sobre a viabilidade de uma arquitetura de

[4]https://www25.senado.leg.br/web/atividade/materias/-/materia/131721

[5]http://www.planalto.gov.br/ccivil_03/leis/l10048.htm

[6]https://www25.senado.leg.br/web/atividade/materias/-/materia/129246

aprendizado profundo para tradução contínua de Libras, almejando uma forma de comunicação dinâmica e bidirecional, considerando que a tradução do português para Libras já foi amplamente pesquisada, com soluções públicas, como VLibras[7] e privadas, como *Hand Talk*[8] disponíveis para a sociedade e, portanto, possibilitar uma maior integração dessa parcela da população a serviços de saúde. Essa arquitetura será unificada em uma arquitetura de aprendizado profundo treinável fim-a-fim e, por fim, a tradução será contínua, já que deverá tratar a dependência temporal e contextual que as línguas naturais apresentam, o que significa que converte uma sequência de sinais em uma sequência de texto.

Na literatura científica, especialmente no contexto brasileiro, estudos envolvendo a tradução automática de Libras baseados em técnicas computacionais não são inéditos; porém, existe uma lacuna de resultados e, consequentemente, soluções maduras o suficiente para serem aplicadas em cenários reais, ou seja, que tragam um benefício tangível para a sociedade e para pessoas surdas, que, muitas vezes, têm enorme dificuldade em ter acesso aos seus direitos fundamentais.

No contexto de serviços médicos, essa barreira é evidenciada no contexto atual de pandemia da Covid-19, causada pelo vírus SARS-CoV-2, em que a demanda por atendimento médico aumentou drasticamente e foi parcialmente convertida em consultas médicas remotas (por exemplo, telessaúde ou telemedicina); portanto, espera-se que uma solução automática de tradução contínua de Libras possa ajudar na comunicação entre um profissional da saúde que não domina Libras e um paciente surdo.

1.2 Tema, Problema e Hipótese de Pesquisa

A presente tese aborda a tradução automática da Língua Brasileira de Sinais no contexto da saúde, com o objetivo de reduzir as barreiras de comunicação enfrentadas por pessoas surdas durante o atendimento médico. Com base na motivação destacada acima e com o objetivo de esclarecer o problema em questão, foi formulada uma questão de pesquisa relacionado a essa tarefa, que resume o problema abordado neste trabalho. O questionamento refere-se à possibilidade de traduzir automaticamente Libras para o português, a partir de um vídeo ou sequência digital de imagens. Finalmente, qual é a eficácia de um tradutor automático em um cenário de vocabulário médico?

Esse questionamento envolve a utilização de técnicas de inteligência artificial para a tradução automática de sentenças em Libras no contexto da saúde, especialmente em situações nas quais um intérprete de Libras não está disponível. O objetivo é desenvolver um processo eficaz, capaz de fornecer traduções precisas e compreensíveis a partir da captação dos sinais através de uma sequência de imagens (vídeo) de uma pessoa executando os gestos, sem a necessidade de dispositivos adicionais, como luvas de captura de movimento ou sensores especializados. A hipótese deste trabalho é investigar se um sistema de tradução automatizada pode ser eficaz ao utilizar apenas a análise visual dos sinais, garantindo precisão e acessibilidade para contextos críticos, como o ambiente de saúde,

[7] https://www.gov.br/governodigital/pt-br/vlibras

[8] https://www.handtalk.me/br/

em que a comunicação precisa e rápida é essencial.

Considerando esses aspectos e visando solucionar o problema em questão, propomos o seguinte questionamento: é possível realizar a tradução automática de Libras utilizando técnicas de inteligência artificial (IA) e visão computacional, caso profissionais de saúde não estejam disponíveis? Assim, este trabalho concentra-se na abordagem desse questionamento, a partir do qual podemos formular a hipótese colocada a seguir.

1.2.1 Hipótese de Pesquisa

A hipótese formulada e discutida ao longo deste trabalho, visando responder os questionamentos de pesquisa previamente apresentados, pode ser expressa da seguinte forma:

É possível realizar a tradução automática da Língua Brasileira de Sinais para o português brasileiro utilizando uma estratégia de tradução contínua baseada em inteligência artificial.

A demonstração e validação da hipótese proposta neste trabalho envolvem a definição e desenvolvimento de um sistema de tradução automática fundamentado na aplicação de técnicas de inteligência artificial, bem como técnicas de processamento de imagens e visão computacional, que possibilita a interpretação dos sinais a partir de um vídeo de uma pessoa executando os gestos. Esse sistema visa realizar a tradução automática de Libras para o português brasileiro, de forma contínua, sem necessitar de intervenção humana, a partir de um vídeo, utilizando uma arquitetura baseada em aprendizado profundo.

1.3 Contribuição

A contribuição única deste trabalho é a proposta de uma solução de tradução automática para sentenças em Língua Brasileira de Sinais no contexto da saúde, particularmente útil em situações nas quais um intérprete profissional não está disponível. O resultado final consiste em um sistema para a tradução automatizada de Libras, com o objetivo de reduzir as barreiras de acesso a serviços de saúde. Este é um problema cotidiano enfrentado pelas pessoas com deficiência auditiva. Para alcançar essa proposta principal, uma série de estudos e desenvolvimentos foi realizada. Esses esforços resultaram em técnicas, metodologias, e outras contribuições parciais, que também podem ser consideradas como sendo resultados relevantes desta tese e foram implementados e testados ao longo do doutorado, destacando-se:

1. Realização de uma extensa revisão da literatura sobre as principais estratégias relacionadas à tradução automática de línguas de sinais para línguas escritas ou orais, possibilitando uma compreensão aprofundada dos desafios e soluções existentes.
2. Concepção e implementação de uma abordagem para a tradução automática contínua de Libras para línguas escritas ou orais, buscando viabilizar um sistema capaz de realizar traduções com precisão e fluidez, essencial no contexto dinâmico dos atendimentos em saúde.
3. Criação e organização de bases de dados específicas para o problema do reconhecimento contínuo de sinais da Língua Brasileira de Sinais, incluindo a coleta e a

anotação de vídeos que representam o uso de Libras no contexto da saúde, garantindo um conjunto de dados robusto e adequado ao treinamento dos modelos de IA.

4. Realização de avaliações quantitativas e qualitativas para medir a acurácia, precisão e capacidade de generalização da solução proposta no reconhecimento contínuo de sinais. Essa avaliação foi essencial para validar a eficácia do sistema, identificando suas limitações e oportunidades de melhoria.

Esses resultados foram fundamentais para a construção da solução principal, além de destacarem a relevância da pesquisa, proporcionando uma tecnologia que promove a inclusão, além de garantir a acessibilidade de pessoas com deficiência auditiva ao sistema de saúde, superando as barreiras comunicacionais de maneira automatizada e eficaz.

1.4 Escopo

Para tentar validar nossa hipótese de pesquisa, este trabalho concentra-se no desenvolvimento de um sistema de tradução automática da Língua Brasileira de Sinais, voltado especificamente para o contexto da saúde. A escolha do domínio de aplicação, restrito ao ambiente de saúde, busca reduzir a complexidade inerente à tradução automática de uma língua visual-espacial para uma língua oral ou escrita. A comunicação em Libras apresenta um vasto repertório de sinais, variações regionais e significados contextuais. Esses fatores tornam o reconhecimento e a tradução um desafio significativo. Assim, ao restringir a aplicação ao contexto da saúde, é possível focar em um conjunto de sinais e expressões que, embora complexos, apresentam maior previsibilidade e especificidade. Essa delimitação reduz a quantidade de variáveis a serem consideradas, facilitando tanto o desenvolvimento de modelos mais precisos quanto a criação de bases de dados específicas que atendam às necessidades do sistema de saúde.

Além disso, a aplicação deste sistema destina-se a situações em que um intérprete de Libras não está disponível e não tem como objetivo substituir o trabalho de um intérprete humano. O sistema foi concebido como um recurso complementar, que busca garantir o acesso à comunicação em circunstâncias em que um intérprete não pode estar presente, contribuindo, assim, para a inclusão e o atendimento adequado das necessidades dos pacientes surdos. Dessa forma, pode ser utilizado em diferentes contextos do setor de saúde, como consultas médicas, atendimentos em hospitais, farmácias e postos de saúde. Isso promove uma comunicação mais acessível e eficaz entre profissionais da saúde e pacientes com deficiência auditiva.

Com essa delimitação do escopo, é possível não só reduzir a complexidade dos dados de entrada e, consequentemente, o custo computacional associado ao treinamento dos modelos, mas também garantir que o sistema atenda com mais eficiência às necessidades críticas de comunicação no ambiente de saúde.

1.5 Estrutura do Texto

Após este Capítulo de introdução, a continuação deste trabalho está estruturada como descrito a seguir. O capítulo 2 apresenta o embasamento teórico, com a fundamentação necessária para o desenvolvimento deste estudo. O capítulo 3 apresenta os trabalhos relacionados com à tradução automática de língua de sinais para línguas orais. O capítulo 4 descreve a solução proposta neste trabalho e os principais componentes. O capítulo 5 descreve os aspectos implementacionais dos métodos utilizados para efetivação do trabalho, incluindo detalhes da base de dados utilizada e as arquiteturas possíveis visionadas. O capítulo 6 apresenta os resultados parciais obtidos até então são exibidos e discutidos. Por fim, o capítulo 7 dá lugar às considerações finais, os problemas encontrados, bem como as limitações do trabalho e um cronograma proposto para o término do trabalho.

Capítulo 2

Embasamento Teórico

Neste capítulo serão apresentados os principais conceitos que fundamentam este trabalho. Inicialmente, serão expostas as principais características, propriedades e conceitos relacionados às línguas de sinais, especialmente a Língua Brasileira de Sinais (Libras). Em seguida, os sistemas de tradução automática serão apresentados, destacando as principais estratégias e métodos utilizados. Por fim, os principais conceitos relacionados ao aprendizado profundo serão apresentados.

2.1 Língua de Sinais

As Línguas de Sinais (LS) são uma forma de comunicação que se desenvolveu como alternativa à fala entre surdos e mudos. Contudo, embora muitas pessoas surdas possuam a capacidade de fala, principalmente aquelas cuja deficiência auditiva foi adquirida após a primeira infância, além de habilidades de comunicação não verbal para se comunicar com pessoas ouvintes, tais métodos de comunicação são geralmente inadequados para comunicação dentro da comunidade surda. Portanto, as mãos se tornaram o principal meio de comunicação dentro dessas comunidades.

As LS são as línguas naturais das comunidades surdas, se configurando como a língua principal e às vezes como a única língua que essa parcela da população usa para comunicações do dia a dia. As Línguas de Sinais são expressas através da combinação de um alfabeto manual, expressões corporais, expressões faciais e movimentos, sendo que a variação de um único componente pode alterar completamente o significado do que se deseja-se comunicar.

A Língua de Sinais tem como meio propagador o campo gesto-visual, o que a diferencia da língua oral, que utiliza o canal oral-auditivo. Além dessa diferença, também apresenta antagonismos quanto às regras constitutivas. No entanto, a língua de sinais deve ser respeitada como língua, pois assume a mesma função da língua oral, a comunicação (Dizeu & Caporali 2005). Assim como as línguas orais, cada país tem sua língua de sinais própria, formada e constantemente se adequando aos seus contextos sócio-culturais, sendo essa forma padrão podendo ainda ser modificada por regionalismos e, com isso, agregando outra característica das línguas naturais, a sua não estaticidade.

Diferente do que muitos imaginam as Línguas de Sinais não são simplesmente mímicas e gestos soltos, utilizados pelos surdos para facilitar a comunicação, qualificação que

só foi desconstruída recentemente a partir dos anos 60, inicialmente através do trabalho de (Stokoe 2005), que argumentou que as línguas de sinais tem status de língua, pois elas são compostas pelos níveis linguísticos: o fonológico, o morfológico, o sintático e o semântico.

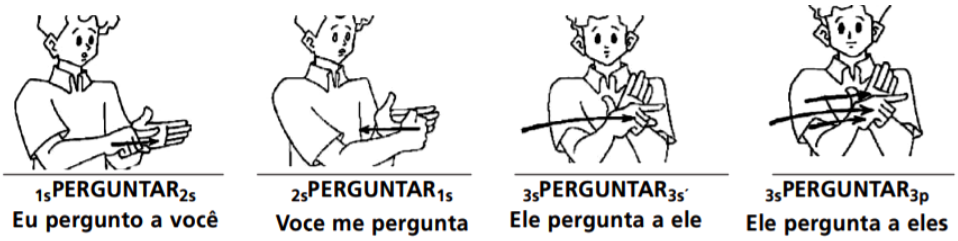

Figura 2.1: Verbos direcionais em Libras. Fonte: (Felipe 2006)

Os sinais são formados a partir da combinação do movimento das mãos com um determinado formato em um determinado lugar, podendo este lugar ser uma parte do corpo ou um espaço em frente ao corpo (Ramos 2004). Além disso, os sinais de uma língua de sinais são compostos por diferentes fonemas, que são unidades básicas ou componentes no qual um sinal pode ser decomposto. Para (Buttussi et al. 2007), cada sinal é unicamente identificado pelos seguintes cinco fonemas:

1. **Configuração de mão**: esse parâmetro representa a posição e forma das mãos e dedos durante a reprodução de um sinal. Para (Felipe 2008), na Libras existem 64 configurações de mão. Cada configuração pode ser executada pela mão dominante ou usando as duas mãos, dependendo do sinal.
2. **Ponto de Articulação**: representa a parte do corpo onde os sinais são realizados, sendo delimitada pela extensão máxima dos braços do emissor, podendo estar localizado em alguma parte do corpo ou em um espaço neutro vertical (do meio do corpo até à cabeça) e horizontal (à frente do emissor).
3. **Movimento**: representa a forma em que a mão é movimentada durante a execução do sinal, embora existam sinais estáticos em algum local e que não necessitam de movimento para serem caracterizados.
4. **Orientação**: representa a orientação ou direção do movimento, ou seja, é o plano em que a palma da mão está orientada.
5. **Expressões não manuais**: são expressões faciais e corporais que podem ser usadas como um traço diferenciador.

Dessa forma, um sinal de Libras é produzido pela combinação desses 4 ou 5 fonemas, sendo importante ressaltar que sinais de uma língua de sinais, como em Libras, com significados distintos, podem ter mais de um componente em comum e serem diferenciados por um único fonema. Esses elementos são conhecidos como articuladores em linguística (Malaia et al. 2018), sendo as mãos os articuladores primários de Libras (Karnopp 1999).

Os surdos possuem uma língua própria não compreendida pela maioria, as línguas de sinais, vivendo assim numa condição de bilinguismo, ou seja, os surdos são minorias

linguísticas por se organizarem em associações onde o fator principal de integração é a utilização de uma língua gestual-visual.

A Língua de Sinais Brasileira (Libras) foi reconhecida através da Lei 10.436[1], de 24 de Abril de 2002, como sendo um meio legal de comunicação e expressão para a comunidade de surdos brasileira. Sendo definida como o sistema linguístico de natureza visual-motora, com estrutura gramatical própria, constituem um sistema linguístico de transmissão de ideias e fatos, oriundos de comunidades de pessoas surdas do Brasil.

Libras é uma língua de fato e direito, portanto, é candidata natural para aplicação de tradução por máquina ou tradução automática, sendo um dos campos mais ativos da área de inteligência artificial. Um artefato de tradução por máquina é um software projetado para realizar a tradução de uma língua fonte para uma língua destino. Apenas recentemente, com o advento do aprendizado profundo, que esses softwares conseguiram atingir padrões de qualidade próximos da tradução humana.

2.2 Redes Neurais Convolucionais

As Redes Neurais Convolucionais (do inglês, *Convolutional Neural Network* - CNN) representam uma das arquiteturas de Aprendizado Profundo (do inglês, *Deep Learning* - DL) mais conhecidas e aplicadas, tendo aplicações nas áreas de visão computacional, processamento de linguagem natural, reconhecimento de padrões, e rastreamento de objetos, entre outras aplicações. As CNNs são particularmente adequadas para o processamento de dados que possuem uma topologia 2D, em formato de grade (matrizes), tais como imagens ou vídeos, e mais recentemente têm sido usadas para tarefas de Processamento de Linguagem Natural (PLN).

Uma rede neural pode ser entendida como sendo um processador distribuído massivamente paralelo, composto de unidades de processamento simples que têm uma propensão natural para armazenar conhecimento experimental e disponibilizá-lo para uso. Os menos céticos dizem que elas "assemelham-se"ao cérebro em dois aspectos fundamentais: **i)** o conhecimento é adquirido pela rede de seu ambiente por meio de um processo de aprendizado ou treinamento. **ii)** as forças de conexão entre os neurônios, conhecidas como pesos sinápticos, são usadas para armazenar o conhecimento adquirido (Haykin 2009).

Sua arquitetura é composta por unidades básicas chamadas de *perceptrons*, um modelo matemático apropriado para representar o neurônio biológico. As entradas do perceptron corresponderiam aos dendritos, que possuem pesos ajustáveis via treinamento, e as saídas, aos axônios. A saída ou processamento do neurônio (Equação 2.1) é computada através da aplicação de uma função de ativação sobre a soma ponderada das entradas do perceptron e do viés com os seus respectivos pesos sinápticos, que gera um valor na saída do neurônio (Haykin 2009).

$$y = \sigma\left(w_0 + \sum_{i=0}^{N} w_i x_i\right) \tag{2.1}$$

[1] http://www.planalto.gov.br/ccivil_03/leis/2002/l10436.htm

As arquiteturas de Redes Neurais Artificiais, considerando interligação e disposição dos neurônios, e ainda a constituição de suas camadas, podem ser divididas em:

- Redes feedforward (alimentação para frente);
- Redes convolucionais;
- Redes recorrentes.

As redes feedforward representam o tipo de rede neural mais básica, sendo caracterizadas por um fluxo unidirecional de dados (alimentação para frente), ou seja, a partir dos dados de entrada, toda a informação é propagada para frente, camada por camada, até atingir a camada de saída da rede. Dependendo do tipo de problema que está sendo modelado, essa arquitetura pode possuir uma topologia simples ou de camadas múltiplas.

Assim como as redes neurais clássicas (Gardner & Dorling 1998), as redes CNNs também são bioinspiradas, tendo seu funcionamento e organização inspirados no córtex visual humano. Neurônios individuais responderiam a estímulos apenas em áreas restritas do campo visual chamadas de campos receptivos. As coleções desses campos se sobrepõem para cobrir toda a área visível. Sendo naturalmente capazes de modelar dependências espaço-temporais através da aplicação de filtros de convolução. Vale lembrar que convolução é também a operação matemática usada para implementar o cômputo de valores de correlação, que alguns cientistas evidenciam que pode ocorrer em forma massiva no córtex cerebral (Marr 1982).

As redes CNN se diferem das redes neurais clássicas *feedforward* pelo uso da operação de convolução ao invés da multiplicação matricial. Nessa topologia de rede, a entrada é processada utilizando campos receptivos locais, aproveitando três ideias importantes que podem melhorar um sistema de aprendizado de máquina: interações esparsas, pesos compartilhados e representações equivariantes. Além disso, uma arquitetura convolucional usual ainda apresenta etapas de subamostragem espacial e normalização, que têm como objetivo reduzir a dimensionalidade espacial das representações e aumentar a capacidade de aprendizado de cada camada individual da rede, respectivamente (LeCun et al. 2015).

A operação de convolução é definida pela Equação 2.2 para uma variável contínua t e pela Equação 2.3 para uma variável discreta t_d.

$$s(t) = (x * w)(t) = \int x(a)w(t-a)da \tag{2.2}$$

$$s(t_d) = (x * w)(t_d) = \sum_{a=-\infty}^{\infty} x(a)w(t_d - a) \tag{2.3}$$

onde x e w são funções reais de t e t_d. No contexto das redes CNNs, é mais comum usar a forma discreta da equação da convolução. A principal aplicação das redes CNNs é para o processamento digital de imagens, portanto, a operação de convolução é expandida para 2 dimensões (2D):

$$S(i,j) = (K * I)(i,j) = \sum_m \sum_n I(i+m, j+n)K(m,n) \tag{2.4}$$

onde $S(i, j)$ denota a convolução do *kernel* K sobre a imagem I, indexado pela linha i e coluna j, m e n indexam a linha e coluna do *kernel* K, respectivamente. Nesta formulação, o rebatimento do *kernel* da operação de convolução é desprezado. Portanto, essa operação é referida algumas vezes na literatura como correlação cruzada.

Além da operação de convolução, um típico bloco convolucional apresenta algumas operações adicionais. Tipicamente, essas operações são camadas de ativação não-lineares e as camadas de agrupamento (Figura 2.2). As camadas de ativação não lineares são camadas que aplicam funções diferenciáveis não-lineares, sendo introduzidas para conferir à rede a capacidade de aproximar relações que aparentemente são não-lineares. Já as camadas de agrupamento ou camadas de subamostragem (do inglês, *Pooling layers*), são responsáveis por realizarem uma redução de dimensionalidade através de uma sumarização estatística do volume de entrada.

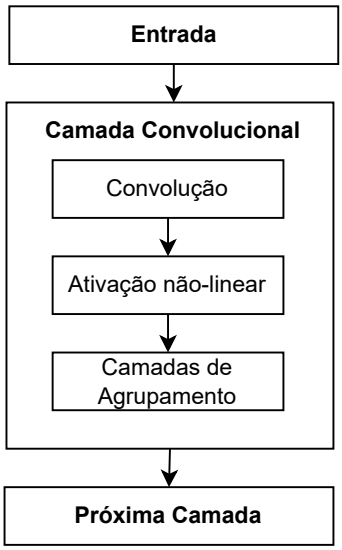

Figura 2.2: Componentes de uma típica rede neural convolucional. Fonte: Autor

Blocos convolucionais produzem mapas de características (do inglês, *feature maps*), que nada mais são do que representações particulares da entrada que enfatizam determinadas partes da entrada. Essas representações são de natureza hierárquica, já que as características são herdadas das camadas posteriores. Essa característica faz com que as redes CNN sejam conhecidas como extratores de características ou codificadores. Ou seja, blocos convolucionais pegam uma imagem como entrada e promovem um processo de redução de dimensionalidade, gerando uma representação (*embedding*) que pode ser usada para alimentar camadas completamente conectadas ou mesmo outras arquiteturas de redes neurais, tais como as redes recorrentes.

2.3 Sequência-para-sequência

Esta é uma classe particular de redes neurais artificiais que é especialmente útil para lidar com sequências, como por exemplo, sequências de texto ou de vídeo. Primeiro, apresentaremos as Redes Neurais Recorrentes e, em seguida, descreveremos a arquitetura mais popular para tradução automática: modelos sequência-para-sequência (do inglês, *sequence to sequence* - (Seq2seq). Apresentaremos também algumas das melhorias que têm sido propostas na literatura e que utilizamos neste trabalho, em particular os chamados mecanismos de atenção.

2.3.1 Redes Neurais Recorrentes

Existem vários tipos de problemas que não podem ser totalmente caracterizados por dados pontuais ou estáticos devido à natureza sequencial desses dados, tipicamente séries temporais, tais como: sinais elétricos, indicadores da bolsa de valores, sentenças de linguagem natural, ou vídeos, todos os quais requerem informações temporais para serem devidamente caracterizados.

Esses tipos de problemas não podem ser modelados por redes neurais clássicas do tipo perceptron multicamadas (PMC ou *Multi Layer Perceptron* - MLP), pois esse tipo de arquitetura não possui a capacidade de reter informações anteriores. Ou seja, são modelos sem memória, e dada a natureza desses problemas, o estado atual tem correlação com pontos anteriores. Logo, fez-se necessário o desenvolvimento de uma nova arquitetura que tivesse a capacidade de contornar esse problema. Para isso, foram desenvolvidas e introduzidas as Redes Neurais Recorrentes, do inglês, *Recurrent Neural Network* - RNN (Goodfellow et al. 2016).

2.3.2 RNN

Uma rede neural recorrente é um modelo baseado em aprendizado profundo para processamento de dados sequenciais. Suas principais características são a existência da propriedade de memória e da capacidade de não requerer entradas e saídas com tamanhos fixos (Sherstinsky 2018) (Figura 2.3). Dado um conjunto de entradas $\mathbf{x^1}, \mathbf{x^2}, \ldots, \mathbf{x^n}$, um conjunto de ativações $\mathbf{a^0}, \mathbf{a^1}, \ldots, \mathbf{a^n}$ e um conjunto de saídas $\mathbf{y^1}, \mathbf{y^2}, \ldots, \mathbf{y^n}$, as equações que governam uma RNN são definidas por:

$$\begin{aligned} a_t &= tanh(W_h a_{t-1} + W_x x_t + b_t) \\ y_t &= W_s a_t \end{aligned} \tag{2.5}$$

onde $\mathbf{W_h}$, $\mathbf{W_x}$ e $\mathbf{W_s}$ são as matrizes de pesos e $\mathbf{b_t}$ é o vetor de vieses. É possível observar que o cálculo da ativação no tempo $\mathbf{a_t}$ também leva em consideração a ativação do tempo $\mathbf{a_{t-1}}$, dando assim a propriedade de memória das RNN, pois todas as informações anteriores também pesarão na predição no tempo $\mathbf{y_t}$. A ativação no tempo $\mathbf{a_0}$ é inicializada com um vetor de zeros.

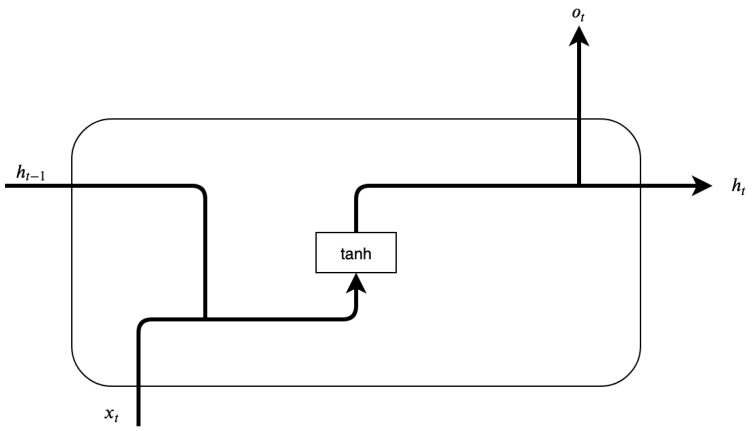

Figura 2.3: Diagrama de uma célula recorrente. Fonte: Autor

Apesar de que, teoricamente, as redes RNN conseguem usar informações passadas para fazer predições no presente, ou seja, possuem memória, na prática, essa capacidade de memorização é profundamente afetada por um problema conhecido como dissipação do gradiente (do inglês, *vanishing gradient problem*). Esse problema, para fins práticos, pode ser visto como a dificuldade da rede em propagar informações. Ou seja, a informação do tempo $\mathbf{t}$ não necessariamente requer afirmações de estados imediatos $\mathbf{t-1}, \mathbf{t-2} \ldots \mathbf{t-10}$, mas sim de estados que já foram apresentados a um tempo considerável $\mathbf{t-k}$ e que provavelmente já foram dissipados durante o processo de treinamento da rede.

Para contornar esse problema foi desenvolvido na literatura um novo tipo de modelo de rede recorrente denominado redes neurais com memória de longo prazo (do inglês, *Long-short Time Memory* - LSTM) tendo como principal característica a capacidade de lidar com longas dependências temporais e, portanto, amenizando o problema de dissipação do gradiente (Goodfellow et al. 2016).

LSTM

Introduzidas por (Hochreiter & Schmidhuber 1997), as redes LSTM têm como sua principal característica a capacidade de processar sequências mais longas e complexas. Essa capacidade é obtida com a adição de portões (do inglês, *gates*) que permitem controlar o fluxo de informação entre as células, permitindo à rede aprender quando manter ou esquecer determinada informação. Uma célula de uma rede LSTM (Figura 2.4) é derivada da célula RNN (Figura 2.3). Porém, introduzindo os conceitos de portão de atualização (do inglês, *update gate*) e portão de esquecimento (do inglês, *forget gate*). Esses portões têm como principal objetivo regular a quantidade de informação que será mantida e passada entre as células.

Dada a sequência $\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_m}$, $\mathbf{h_{t-1}}$ e $\mathbf{c_{t-1}}$, onde $\mathbf{m}$ é o comprimento da sequência, $\mathbf{x_i} \in \mathbb{R}^{\mathbf{d}}$ é o vetor obtido pela concatenação de características, $\mathbf{h_{t-1}}$ e $\mathbf{c_{t-1}}$ são os estados ocultos e o estado anterior da célula LSTM ($\mathbf{h_o}$ e $\mathbf{c_o}$ são iniciados com vetores com zeros),

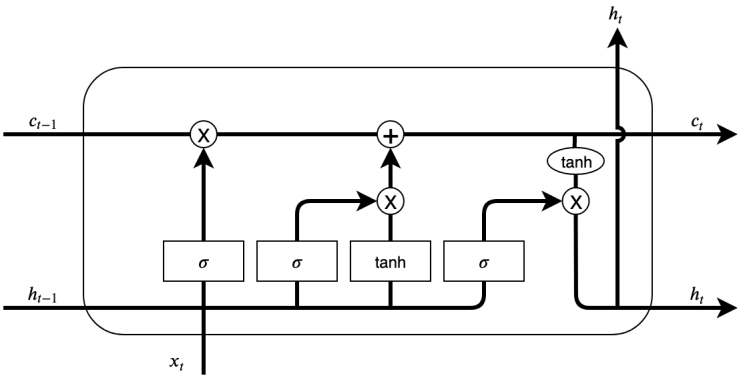

Figura 2.4: Diagrama de uma célula LSTM. Fonte: Autor

o novo estado oculto e novo estado da célula são computados pelo uso das seguintes expressões:

$$\begin{aligned} \hat{c}_t &= tanh(W_c[h_{t_i}, x_t] + b_c) \\ i_t &= \sigma(W_i[h_{t_i}, x_t] + b_i) \\ f_t &= \sigma(W_f[h_{t_i}, x_t] + b_f) \\ o_t &= \sigma(W_o[h_{t_i}, x_t] + b_o) \\ c_t &= i_t * \hat{c}_t + f_t * c_{t-1} \\ h_t &= tanh(c_t) * o_t \end{aligned} \tag{2.6}$$

sendo σ a função de ativação sigmoide e $*$ denotando o produto de Hadamard. $\mathbf{W_f}$, $\mathbf{W_i}$, $\mathbf{W_o}$, $\mathbf{W_c} \in \mathbb{R}^{(\mathbf{N}+\mathbf{d})\times\mathbf{N}}$ são matrizes de pesos e $\mathbf{b_f}$, $\mathbf{b_i}$, $\mathbf{b_o}$, $\mathbf{b_c} \in \mathbb{R}^{\mathbf{N}}$ são os vetores de vieses. As matrizes de pesos e vetores de vieses são inicializados randomicamente e são atualizadas (aprendem) por um processo de otimização durante a fase de treinamento, $\mathbf{N}$ é o tamanho da camada LSTM e $\mathbf{d}$ é a dimensão do vetor de características de entrada.

GRU

As redes do tipo Unidade Recorrente Fechada (do inglês, *Gated Recurrent Neural Networks* - GRU) (Chung et al. 2014), são uma alternativa às redes LSTM. Esse tipo de rede é bastante similar à rede LSTM, sendo que os portões de entrada $\mathbf{i_t}$ e de esquecimento $\mathbf{f_t}$ são unificados em um único portão de atualização $\mathbf{z_t}$. Já o portão de saída $\mathbf{o_t}$ é removido e substituído por um portão de reinicialização (*reset*) $\mathbf{r_t}$:

$$\begin{aligned} z_t &= \sigma(W_z[h_{t_z}, x_t] + b_z) \\ r_t &= \sigma(W_r[h_{t_i}, x_t] + b_r) \\ h_t &= z_t * h_{t-1} + (1 - z_t) * tanh(W_h[r_t * h_{t_i}, x_t] + b_h) \end{aligned} \tag{2.7}$$

sendo σ a função de ativação sigmoide e $*$ denotando o produto de Hadamard. $\mathbf{W_z}$, $\mathbf{W_r}$,

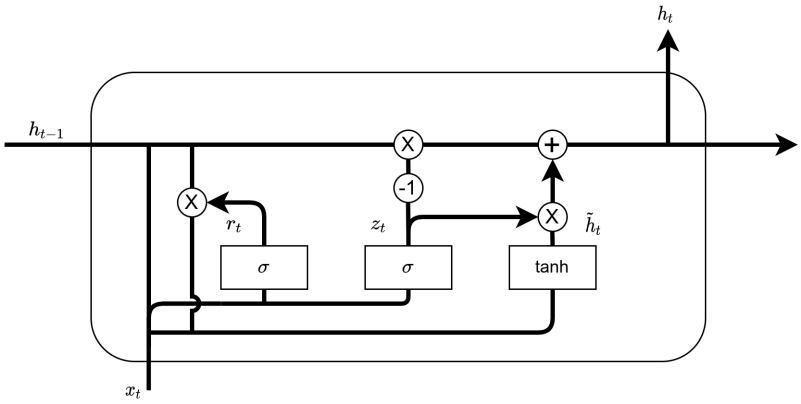

Figura 2.5: Diagrama de uma célula GRU. Fonte: Autor

$\mathbf{W_h} \in \mathbb{R}^{(\mathbf{N}+\mathbf{d})\times\mathbf{N}}$ são matrizes de pesos e $\mathbf{b_z}$, $\mathbf{b_r}$, $\mathbf{b_h} \in \mathbb{R}^{\mathbf{N}}$ são vetores de vieses. Matrizes de pesos e vetores de vieses são inicializados randomicamente e aprendem por um processo de otimização durante a fase de treinamento, **N** é o tamanho da camada GRU e **d** é a dimensão do vetor de características de entrada.

Uma rede GRU também é caracterizada por ter como saída seu próprio estado oculto $\mathbf{h_t}$ (ver Figura 2.5). Uma consequência natural do menor número de portões são um menor número de parâmetros treináveis, portanto, apesar de apresentaram resultados parecidos, GRU são, no geral, mais fáceis de serem treinadas, ou seja, apresentam um menor custo computacional.

As redes LSTM ou GRU são uma das arquiteturas de aprendizado profundo mais usadas para fins práticos, sendo uma das principais técnicas utilizadas na área de Processamento Língua Linguagem Natural e um dos principais fatores que trouxeram ganhos expressivos na qualidade de serviços de tradução automática de texto. E, mais recentemente, passou a ter grande uso em modelos de classificação de ações ou descrição de cenas a partir de vídeo através das arquiteturas sequência-para-sequência (do inglês, *sequence-to-sequence* - Seq2Seq).

2.3.3 Modelo Sequência-para-sequência

Os modelos de sequência a sequência (seq2seq)(Sutskever et al. 2014) mais comuns são os modelos compostos de um codificador (do inglês, *encoder*) e um decodificador (do inglês, *decoder*), que geralmente usam uma rede neural recorrente (RNN) para codificar a sentença de origem (entrada) em um único vetor, usualmente denominado na literatura como vetor de contexto. Podemos pensar no vetor de contexto como sendo uma representação abstrata de toda a sentença de entrada. Este vetor é então decodificado por uma segunda RNN que aprende a reconstruir a sentença alvo (saída) gerando uma palavra de cada vez. Uma representação usual de uma arquitetura Seq2Seq pode ser vista a seguir:

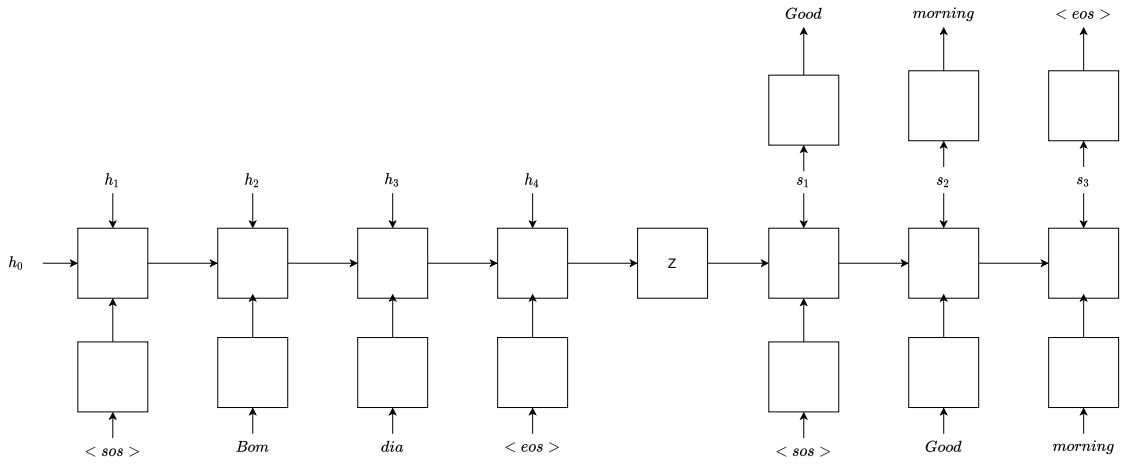

Figura 2.6: Modelo Seq2Seq. Fonte: Autor

Descrição

O conjunto de treinamento consiste em uma lista de pares $(\mathbf{x}, \mathbf{z})$, onde $\mathbf{x} = \mathbf{x_1}, \ldots, \mathbf{x_T}$ é uma sequência origem de comprimento $\mathbf{T}$; e $\mathbf{y} = \mathbf{y_1}, \ldots, \mathbf{y_T}$ é uma sequência destino de comprimento. Dado $\mathbf{x}$ aprendemos para prever $\mathbf{y}$.

Os vetores $\mathbf{V_x}$ e $\mathbf{V_y}$ são os vocabulários de origem e destino, e $|\mathbf{V}|$ e $|\mathbf{V_y}|$ seu respectivo tamanho. $\mathbf{E_x} \in \mathbb{R}^{|\mathbf{V_x}| \times \mathbf{m}}$ e $\mathbf{E_y} \in \mathbb{R}^{|\mathbf{V_y}| \times \mathbf{m}}$ são as matrizes de *embeddings*[2] das sequências de origem e destino. Essas matrizes são inicializadas aleatoriamente e treinadas em conjunto com os demais parâmetros do modelo.

Codificador (*Encoder*)

O bloco de codificação é usualmente formado por uma rede recorrente, sendo mais comum o uso de redes LSTM ou GRU, já que essas duas classes de redes recorrentes são mais resilientes ao problema da dissipação do gradiente.

A rede recorrente do codificador recebe a sequência $\mathbf{h(t)}$ e produz outro vetor de contexto denominado $\mathbf{z(t)}$. A cada passo de tempo, a entrada para o codificador é o *embedding* da entrada atual $\mathbf{e(x_t)}$, bem como o estado oculto do passo de tempo anterior $\mathbf{h_{t-1}}$, e o codificador emite um novo estado oculto $\mathbf{h_t}$. Podemos pensar no estado oculto como uma representação vetorial do vídeo até o instante de tempo $\mathbf{t}$. Portanto, podemos representar o codificador como:

$$h_t = encoder(e(x_t), h_{t-1}) \tag{2.8}$$

[2]Projeção de um vetor de alta dimensão (esparso) em um vetor de baixa dimensão.

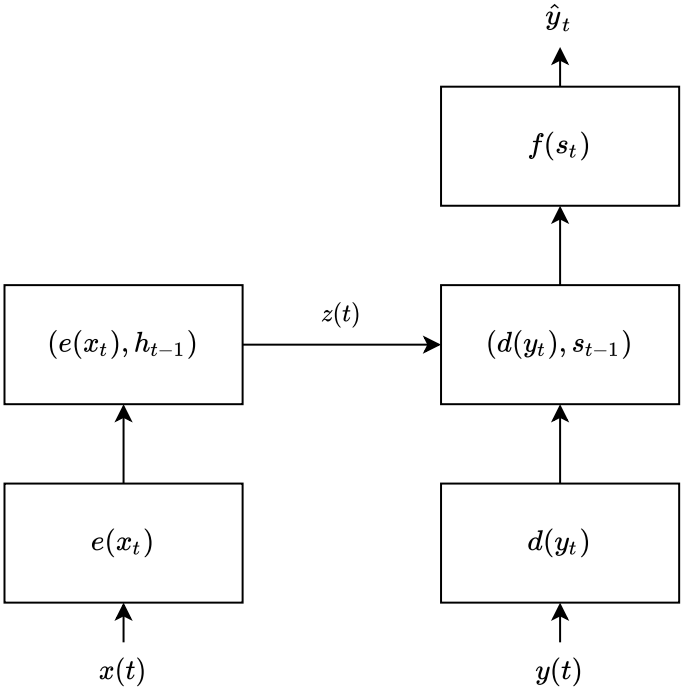

Figura 2.7: Fluxo de transformações do modelo Seq2Seq. Fonte: Autor

Decodificador (*Decoder*)

O bloco do decodificador é formado por outra rede recorrente, usualmente do mesmo tipo usado pelo codificador. De posse do vetor de contexto $\mathbf{h_t} = \mathbf{z}$, podemos começar a decodificá-lo para obter a sentença destino (tradução). A cada passo de tempo **t**, a entrada para o decodificador é o *embedding* **d**, da palavra atual $\mathbf{d}(\mathbf{y_t})$, bem como o estado oculto do passo de tempo anterior $\mathbf{s_{t-1}}$, onde o estado oculto inicial do decodificador $\mathbf{s_0}$, é o vetor de contexto **z**, ou seja, o estado oculto inicial do decodificador é o estado oculto do codificador final $\mathbf{s_0} = \mathbf{z} = \mathbf{h_t}$. Assim, semelhante ao codificador, podemos representar o decodificador como:

$$s_t = decoder(d(y_t), s_{t-1}) \tag{2.9}$$

No decodificador, precisamos ir do estado oculto **s** para uma palavra real, portanto, a cada passo de tempo **t**, usamos para prever, o que pensamos ser a próxima palavra na sequência, para isso é usada uma camada linear **f**:

$$\hat{y}_t = f(s_t) \tag{2.10}$$

Função objetivo

A função da rede recorrente é estimar a probabilidade condicional $\mathbf{p}(\mathbf{y_1},\ldots,\mathbf{y_{T'}}|\mathbf{x_1},\ldots,\mathbf{x_T})$, onde $\mathbf{x_1},\ldots,\mathbf{x_t}$ é a sequência de entrada e $\mathbf{y_1},\ldots,\mathbf{y_t}$ é a sequência de saída, os comprimentos da sequência de entrada $\mathbf{T'}$ e da sequência de saída $\mathbf{T}$ podem ser diferentes, portanto, o modelo pode ser formalizado como:

$$P(y/x) = \prod_{t}^{T'} p(\hat{y_t} = y_t|X) \tag{2.11}$$

Sendo cada distribuição $\mathbf{p}(\hat{\mathbf{y}}_\mathbf{t} = \mathbf{y_t}|\mathbf{X})$ representada com um *softmax* sobre todos os *tokens* do vocabulário. O processo de otimização de uma rede Seq2Seq é feito através da minimização do logaritmo da probabilidade de uma tradução correta $\mathbf{y}$ dado uma sequência de origem $\mathbf{x}$:

$$loss(x,y) = -log \cdot P(y|x) = -\sum_{i=1}^{T'} logp(\hat{z}_t = z_t|x) \tag{2.12}$$

$$\mathcal{L} = \frac{1}{\mathcal{D}} \sum_{x,y \in \mathcal{D}} loss(x,y) \tag{2.13}$$

onde $\mathcal{D}$ é o *batch* de treinamento. No processo de inferência, as traduções são produzidas a partir da tradução mais provável, sendo dada por:

$$\hat{y} = arg\max_{y} p(y|x) \tag{2.14}$$

Apesar da arquitetura Seq2seq ter resultados expressivos para tradução automática ela apresenta um problema conhecido como *bottleneck problem*. Esse problema é causado pela necessidade de capturar toda a informação da entrada em um único vetor de contexto $\mathbf{h_t} = \mathbf{z}$ de tamanho fixo, conforme o tamanho da entrada aumenta, fica mais difícil comprimir toda a informação importante da entrada nesse vetor. Consequentemente, seu desempenho diminui com frases longas, pois o modelo acaba esquecendo partes importantes. Esse problema pode ser endereçado dotando o modelo com a capacidade de enxergar todos os estados ocultos a qualquer momento, sendo essa a ideia básica dos mecanismos de atenção.

Mecanismos de Atenção

O conceito de mecanismo de atenção foi introduzido por Bahdanau et al. (2015) e se tornou um conceito predominante na literatura das redes neurais, sendo um componente essencial em diversas arquiteturas que vieram a surgir. Primeiro, na área de PLN (Vaswani et al. 2017, Radford et al. 2019*a*) e posteriormente na área de visão computacional (Xu et al. 2015, Hu et al. 2017). Assim como as redes neurais, o conceito de atenção tam-

bém tem inspiração biológica, já que o nosso sistema visual tende a focar seletivamente em algumas partes da imagem, enquanto ignora informações irrelevantes para algum determinado contexto. Por exemplo, durante a execução de um sinal de libras, as mãos e o rosto recebem mais atenção, pois como visto na Seção 2.1, estes são os dois grandes elementos de um sinal.

O modelo de atenção proposto por Bahdanau et al. (2015) foi concebido tendo a arquitetura Seq2Seq como contexto. Basicamente, uma rede recorrente (codificador) pega uma sequência de *tokens* $\mathbf{x} = \mathbf{x_1}, \ldots, \mathbf{x_T}$ e faz a codificação dessa entrada em um vetor de contexto $\mathbf{h} = \mathbf{h_1}, \ldots, \mathbf{h_T}$, que por sua vez é decodificado por outra rede recorrente (decodificador) em uma sequência de saída $\mathbf{y} = \mathbf{y_1}, \ldots, \mathbf{y_T}$. Em cada posição de tempo $\mathbf{t}$, $\mathbf{h_t}$ e $\mathbf{s_t}$ são os estados ocultos do codificador e decodificador, respectivamente.

Nesse contexto, o mecanismo de atenção permite o decodificador acessar toda sequência codificada $\mathbf{h} = \mathbf{h_1}, \ldots, \mathbf{h_T}$. Em outras palavras, a sequência codificada é ponderada por um vetor de contexto $\mathbf{c_i}$, priorizando posições em $\mathbf{h}$ onde existe informação mais relevante para predição do próximo *token* da sentença. O vetor de contexto $\mathbf{c_i}$ pode ser definido através da fórmula:

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j \tag{2.15}$$

A distribuição de atenção α_{ij} de cada posição h_j é calculado como:

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{T_x} exp(e_{ik})} \tag{2.16}$$

onde,

$$e_{ij} = v_a^T tanh(W_a s_{i-1}, U_a h_j) \tag{2.17}$$

Além do mecanismo de atenção aditivo, também existe um mecanismo de atenção de caráter multiplicativo proposto por (Luong et al. 2015). Sua principal diferença é o uso de uma função agregativa multiplicativa $\mathbf{e_{ij}}$:

$$e_{ij} = s_i^T W_a \bar{h}_j \tag{2.18}$$

Mecanismos de atenção podem ser generalizados (Chaudhari et al. 2021). Para isso, a atenção pode ser vista como o processo de mapeamento de uma sequência de chaves, ou *keys* ($\mathbf{K}$), para uma distribuição de atenção α de acordo com uma consulta, ou *query* ($\mathbf{q}$), onde as chaves são os estados ocultos $\mathbf{h_i}$ e uma consulta (*query*) é um único estado oculto do decodificador $\mathbf{s_{j-1}}$. Nesta formulação, a distribuição de atenção $\alpha_{\mathbf{ij}}$ enfatiza as chaves que são relevantes para a tarefa principal em relação à consulta $\mathbf{q}$ como:

$$A(q,K,V) = \sum_i p(a(k_i,q) * v_i \tag{2.19}$$

A variável **a** indica a função de alinhamento e **p** a função de distribuição, sendo responsáveis por indicar como as chaves e a consulta são combinadas para produzir pesos de atenção. A função de alinhamento é uma rede *feedforward*, sendo responsável por aprender os pesos de atenção α_{ij} em função dos estados $\mathbf{h_i}$ e $\mathbf{s_j}$ e produzir escores de energia $\mathbf{e_{ij}}$. A função distribuição, por sua vez, converte os escores de energia em pesos de atenção, sendo **a** e **p** função diferenciáveis. Ao longo do tempo, foram introduzidas diversas formulações diferentes de funções de alinhamento **a** e de **p**, algumas destas projetadas para cenários e tarefas específicas.

Para algumas tarefas, por exemplo, classificação de texto ou análise de sentimento, a entrada é uma sequência de *tokens* $\mathbf{x} = \mathbf{x_1}, \ldots, \mathbf{x_T}$. Porém, a saída não é uma sequência, podendo ser uma classe ou valor numérico. Nesse contexto, o mecanismo de atenção pode ser usado para aprender quais são os *tokens* mais importantes dessa sequência. Para isso, Yang et al. (2016) propõe o modelo de auto-atenção (no inglês, *self-attention*), chamada também de intra-atenção, sendo a rede caracterizado por relacionar diferentes posições (*tokens*) de uma única sequência. Ou seja, as consultas **q** e as chaves **k** pertencem à mesma sequência. Este é um dos principais conceitos de outra arquitetura do tipo Seq2Seq, que é a arquitetura conhecida como *Transformers* (Vaswani et al. 2017).

2.3.4 Redes Neurais Transformadoras

Antes de Vaswani et al. (2017) introduzirem as Redes Neurais Transformadoras (do inglês, *Transformer Neural Network*), as redes recorrentes foram o bloco fundamental das arquiteturas Seq2Seq. Porém, essa configuração herda um problema crônico que atinge as redes recorrentes: o *vanishing gradient problem*, que torna difícil lidar com dependências temporais longas, exigindo o uso conjunto de redes recorrentes (LSTM ou GRU) e mecanismos de atenção. Nesse contexto, a premissa básica que motivou a concepção das Redes Neurais Transformadoras (RNT) é que se os mecanismos de atenção podem acessar qualquer estado oculto, então as redes recorrentes podem ser removidas e, em vez disso, contando inteiramente com um mecanismo de atenção para modelar as dependências globais entre entrada e a saída.

Mecanismo de *Self-Attention Multi-Head*

Na rede neural do tipo *transformer*, não existe apenas um único bloco de atenção. No seu lugar, existe um mecanismo denominado Mecanismo de Auto-atenção Multi-cabeça (no inglês, *Multi-Head Attention*), que computa a atenção considerando vários aspectos distintos da entrada. Ou seja, cada "cabeça" tem sua opinião, enquanto o processo de decisão é realizada através de um voto equilibrado de todas as "cabeças" do modelo *transformer* (ver Figura 2.9).

Para aprender representações diversas da entrada, cada "cabeça" é uma transformação

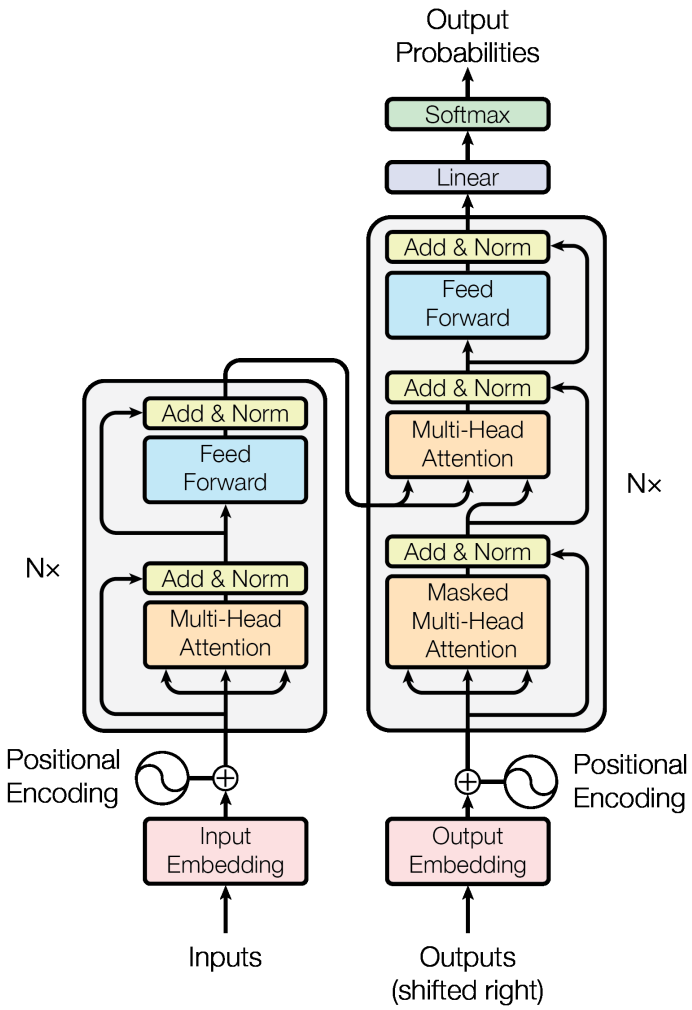

Figura 2.8: Redes Neurais *Transformers*. Fonte:Vaswani et al. (2017)

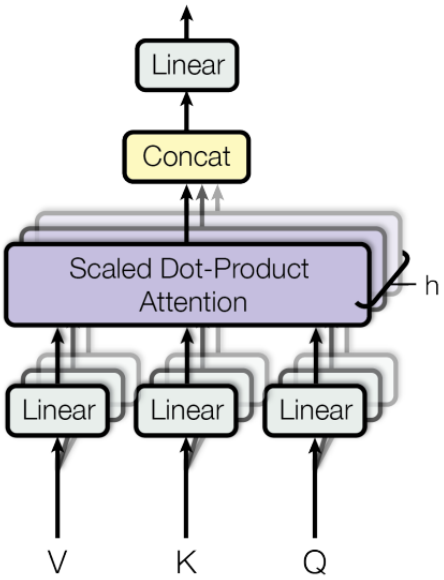

Figura 2.9: Mecanismo *Self-attention Mmulti-head*. Fonte:(Vaswani et al. 2017)

linear única da representação da entrada visto como um conjunto de matrizes: **Q**, **K** e **V**. A matriz **Q** é uma matriz que contém um conjunto de consultas (*query*), **K** é uma matriz de chaves (*key*), **V** é a matriz de valores (*value*) e **d** é a dimensão do vetor chave. No contexto de uma arquitetura Seq2seq, as matrizes **Q** e **V** fazem o papel do estado oculto (*hidden state*) do codificador e decodificador, respectivamente.

$$Attention(Q,K,V) = \left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.20}$$

Codificação Posicional

Ao contrário das redes recorrentes ou mesmo das redes convolucionais, as redes transformadoras não possuem a habilidade natural de modelar a ordem da sequência original (informação posicional). A informação de posição e a ordem de um elemento em uma sequência são fundamentais para modelagem de linguagens, já que a mesma é usada para representar relações gramaticais ou mesmo semânticas entre elementos de uma sentença. Para isso, é introduzido o conceito de Codificação Posicional (do inglês, *Positional Encoding* - PE), que basicamente consiste na introdução de informação sobre a posição relativa ou absoluta dos *tokens* da sequência original:

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{model}}) \tag{2.21}$$

$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{model}}) \tag{2.22}$$

onde **pos** é a posição e **i** é a dimensão. Ou seja, cada dimensão da codificação posicional

corresponde a uma senoide. Os comprimentos de onda formam uma progressão geométrica de $\mathbf{2\pi}$ a $\mathbf{100002\pi}$. Portanto, a codificação posicional pode ser vista como um vetor $\mathbf{\tilde{p}_t}$ contendo pares de senos e cossenos para cada frequência, tendo a mesma dimensão de *embedding* do modelo $\mathbf{d_{model}}$.

$$\vec{p}_t = \begin{bmatrix} sin(\omega_1 t) \\ cos(\omega_1 t) \\ \vdots \\ sin(\omega_d/2t) \\ cos(\omega_d/2t) \end{bmatrix}_{dx1} \tag{2.23}$$

Um efeito colateral da remoção das redes recorrentes e suas recorrências é a capacidade de uma melhor paralelização, ou seja, as RNT tem um desempenho computacional muito superior em um ambiente de computação massivamente paralelo baseado em processadores gráficos (GPUs).

As Redes Neurais *Transformers* (RNT) tornaram-se o estado da arte em diversas tarefas de processamento de linguagem natural. Sendo o curso natural a expansão do conceito de redes baseadas inteiramente em mecanismos de atenção para outras áreas da Inteligência Artificial, em especial, para a área de Visão Computacional, surgiram, então, uma classe de redes denominada Transformadores Visuais (no inglês, *Vision Transformers* - ViT). Essas redes são caracterizadas pela substituição dos blocos convolucionais por blocos baseados em mecanismos de atenção.

2.3.5 Vision Transformers

Introduzida primeiramente por Dosovitskiy et al. (2020), a arquitetura *Vision Transformers* pouco difere das RNT. Sua única grande diferença é a introdução do conceito de *patch embedding*, que nada mais é que uma transformação necessária para alimentar uma rede RNT usando uma imagem como entrada. Para isso, uma imagem $\mathbf{x} \in \mathbb{R}^{\mathbf{H}\times\mathbf{W}\times\mathbf{C}}$ é convertida em um sequência unidimensional (1D) de *patches* $\mathbf{x_p} \in \mathbb{R}^{\mathbf{N}\times(\mathbf{P^2C})}$, sendo $(\mathbf{H},\mathbf{W})$ a resolução da imagem, $\mathbf{C}$ o número de canais, $(\mathbf{P},\mathbf{P})$ a resolução de cada *patch* da imagem, e $\mathbf{N} = \mathbf{HW}/\mathbf{P^2}$ o número resultante de *patches*, servindo como o comprimento efetivo da sequência de entrada da RNT. A rede transformadora usa um vetor latente de tamanho constante $\mathbf{D}$ em todas as suas camadas. Então, os *patches* são convertidos em um sequência unidimensional (1D) e mapeados para $\mathbf{D}$ dimensões, através de uma projeção linear treinável (Eq. 2.24 a 2.26).

$$\mathbf{z_0} = \left[x_{class}; x_p^1\mathbf{E}; x_p^2\mathbf{E}; \ldots; x_p^N\mathbf{E}\right] + \mathbf{E}_{pos}, \quad \mathbf{E} \in \mathbb{R}^{(P^2C)\times D}, \mathbf{E}_{pos} \in \mathbb{R}^{(N+1)\times D} \tag{2.24}$$

$$\mathbf{z}_\ell^{'} = MSA(LN(\mathbf{z}_{\ell-\mathbf{1}})) + \mathbf{z}_{\ell-\mathbf{1}}, \quad \ell = 1\ldots L \tag{2.25}$$

$$\mathbf{z}_\ell = MLP(LN(\mathbf{z}_\ell^{'})) + \mathbf{z}_\ell^{'}, \quad \ell = 1\ldots L \tag{2.26}$$

$$\mathbf{y} = LN(\mathbf{z_L^0}) \tag{2.27}$$

A rede RNT, por padrão, é uma arquitetura Seq2Seq, ou seja, uma sequência de entrada é codificada por um codificador, que por sua vez é decodificada em uma sequência de saída por um decodificador, porém, a rede ViT é composta apenas de um codificador (Ver Figura 2.10). Portanto, o comprimento da sequência de entrada é igual ao comprimento da sequência de saída (número de *patches*). Porém, não é viável aplicar uma camada de classificação na sequência de saída. Para isso, é introduzido um parâmetro fictício $\mathbf{x}_{class}$, conhecido também como *token class*. O *token class* é apenas um número que é colocado em cada sequência $\mathbf{z}_0^0 = \mathbf{x}_{class}$, usado para representar o entendimento $\mathbf{y}$ (Eq. 2.29) sobre toda a imagem (de *patches* projetados).

Até agora, o modelo não tem informação sobre a posição original dos *patches*. Precisamos passar essa codificação posicional $\mathbf{x_{pe}}$, para isso é introduzido parâmetro treinável na forma de vetor unidimensional $\mathbf{x_{pe}} \in \mathbb{R}^{(\mathbf{N+1})\mathbf{xd_{size}}}$. Por fim, o codificador da RNT consiste em camadas alternadas de *self-attention multi-head* MSA (Eq. 2.27) e perceptron multicamadas MLP (Eq. 2.28). Normalização de camadas (ou *layers normalization*) (LN) é aplicada antes de cada bloco, e conexões residuais são aplicadas após cada bloco.

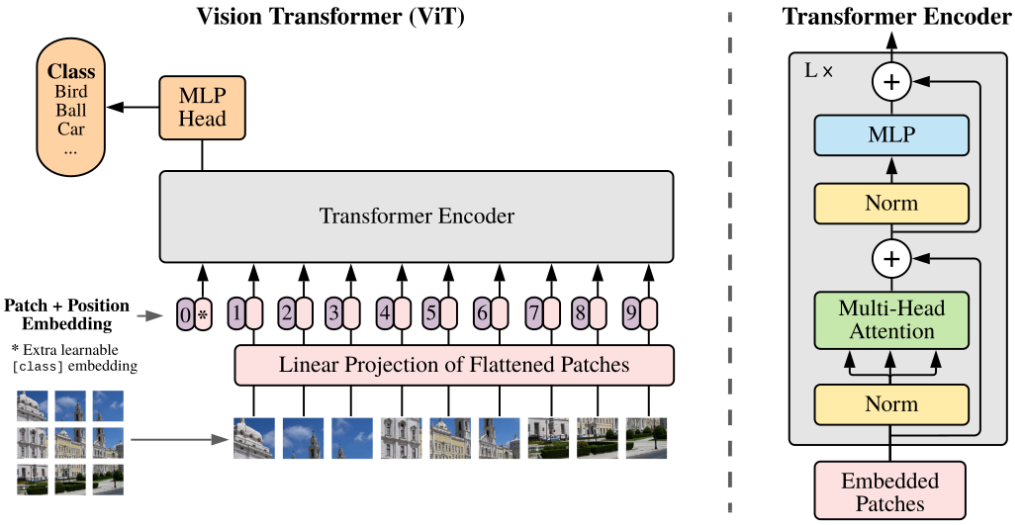

Figura 2.10: Rede Neural *Vision Transformer*. Fonte: Dosovitskiy et al. (2020)

2.3.6 Função de custo

As funções de custo são usadas para aprender os parâmetros que melhor explicam o conjunto de dados disponíveis (amostra), sendo um dos aspectos mais importantes no projeto de uma rede neural. A maioria das redes neurais modernas são treinadas usando a máxima verossimilhança. Isso significa que a função de custo é simplesmente a probabilidade logarítmica negativa (Goodfellow et al. 2016). Esta função de custo é pode ser definida por:

$$J(\theta) = -\mathbb{E}_{x,y\sim\hat{p}_{data}} log\, p_{model}(y|x) \tag{2.28}$$

Equivalentemente descrita como a entropia cruzada entre os dados de treinamento e a distribuição do modelo:

$$CE(p|t) = -\sum_i t_i \cdot log p_i \tag{2.29}$$

onde t_i é a distribuição de probabilidade desejada (dados) e p_i é a distribuição de probabilidade obtida, computada com a função softmax (modelo).

Outra função bastante usada é o erro médio quadrático:

$$MSE = \frac{1}{n}\sum_{i=}^{n}(y_i - \hat{y}_i)^2 \tag{2.30}$$

A função de custo total usada para treinar uma rede neural geralmente apresenta um termo de regularização.

2.3.7 Regularização

Métodos de aprendizado profundo aprendem a partir de dados e um dos maiores problemas que esses métodos encontram é o problema de sobre-ajuste, como o próprio nome diz, esse problema pode ser caracterizado como um sobre-ajuste do modelo aos dados de treinamento, não sendo capaz de generalizar para novas amostras que não foram vistas pelo modelo durante o treinamento. Esse problema é abordado através de métodos de regularização.

Nesse contexto, regularização é definida como qualquer modificação que fazemos em um algoritmo de aprendizado que visa reduzir seu erro de generalização, mas não seu erro de treinamento (Goodfellow et al. 2016). Muitas das abordagens de regularização são baseadas em um processo de penalização aplicado sobre a função objetivo do modelo. Uma função objetivo regularizada pode ser definida como:

$$\tilde{J}(\theta;X,y) = J(\theta;X,y) + \alpha\Omega(\theta) \tag{2.31}$$

Onde $\alpha \in [\mathbf{0},\infty)$ é um hiperparâmetro que pondera a contribuição relativa do termo de penalidade da norma, Ω, em relação à função objetivo padrão $\tilde{\mathbf{J}}(\mathbf{x},\theta)$. Definir α como 0 resulta em nenhuma regularização. Valores maiores de α correspondem a mais regularização (Goodfellow et al. 2016). Uma das penalidades mais usadas é conhecida como L1 ou Lasso, onde $\Omega(\theta)$ é a soma dos valores absolutos dos parâmetros $||w||_1$:

$$\tilde{J}(\theta;X,y) = J(\theta;X,y) + \alpha||w||_1 \tag{2.32}$$

Outra penalidade bastante usada é a norma L2 ou Ridge, onde a penalidade $\Omega(\theta)$ é a soma dos quadrados dos parâmetros $\frac{1}{2}||w||_2^2$:

$$\tilde{J}(\theta;X,y) = J(\theta;X,y) + \alpha\frac{1}{2}||w||_2^2 \tag{2.33}$$

Sendo a regularização L2 mais usada na regularização de redes neurais, neste contexto, ela é também conhecida como decaimento dos pesos, já que a mesma torna os pesos cada vez menores e próximos a zero. Uma outra estratégia de regularização própria das redes neurais é conhecida como *Dropout* (Srivastava et al. 2014), onde durante o treinamento da rede, camadas ocultas são desligadas aleatoriamente com uma probabilidade **p**, de maneira temporária, ou seja, a complexidade do modelo é temporariamente reduzida, tornando o treinamento mais robusto.

Formalmente, suponha que um vetor de μ especifique quais unidades incluir, e $J(\theta,\mu)$ define o custo do modelo definido pelos parâmetros θ e máscara μ. Então o treinamento de *dropout* consiste em minimizar $\mathbb{E}_\mu J(\theta,\mu)$. O valor esperado contém muitos termos, mas podemos obter uma estimativa não enviesada do seu gradiente por valores amostrais de μ (Goodfellow et al. 2016).

2.4 Considerações

Neste Capítulo, discorremos sobre a fundamentação teórica necessária para a caracterização da problemática de estudo. Isto é, sobre todos os elementos que serão tratados no âmbito do presente trabalho, seja sobre a temática de estudo em si, ou sobre os demais elementos que fazem parte da metodologia, que será apresentada no Capítulo 5.

Como esta tese tem sua metodologia fundamentada em técnicas ou modelos de reconhecimento de ações baseados em aprendizado profundo, procurou-se fazer uma revisão sucinta das principais abordagens que obtiveram sucesso e podem ser aplicadas ao problema de tradução automática contínua de sentenças de Libras.

No próximo capítulo, será apresentada uma seleção de trabalhos relacionados ao tema deste trabalho, com o objetivo de mostrar o estado da arte. Serão abordadas as diferentes técnicas de aprendizado de máquina que já foram aplicadas na resolução desse problema, bem como trabalhos que serviram de inspiração para a metodologia que será apresentada no 5 desta tese.

Capítulo 3

Trabalhos Relacionados

Um grande número de trabalhos sobre aprendizado profundo voltado para acessibilidade vem sendo desenvolvido nos últimos anos. Entre esses, o reconhecimento e a tradução automática de Língua de Sinais se destacam como uma das áreas de pesquisa mais ativas e desafiadoras. Nesse contexto, são apresentados alguns trabalhos relevantes que situam e caracterizam o estado da arte da pesquisa.

3.1 Contextualização no Estado da Arte

O problema de reconhecimento automático das Línguas de Sinais (do inglês, *Sign language recognition* - SLR) tem como objetivo a tradução de sinais a partir de uma sequência de imagens digitais (vídeos) em um conjunto ordenado de glosas de sinais. Tradicionalmente, os métodos de SLR são subdivididos em duas categorias distintas: reconhecimento de língua de sinais isolado (do inglês, *Isolated Sign Language Recognition* - ISLR) (Huang et al. 2019, Zhang et al. 2016, Yin et al. 2016, Koller et al. 2016) e reconhecimento de língua de sinais contínuo (do inglês, *Continuous Sign Language recognition* - CSRL) (Huang et al. 2018, Guo et al. 2018, Camgoz et al. 2018*b*, Stoll et al. 2018), sendo que o primeiro faz a tradução de um sinal para uma palavra, e o segundo faz a tradução de um conjunto de sinais em uma sentença, tendo assim uma maior aplicabilidade prática (Wei et al. 2019).

As arquiteturas para CSRL são usualmente caracterizadas pelo acoplamento de Redes Neurais Convolucionais (LeCun et al. 2015) que servem como codificadores fazendo a compressão da dimensão de cada quadro do vídeo. Por sua vez, a saída das redes normalmente alimenta Redes Recorrentes, tais como as LSTM (Hochreiter & Schmidhuber 1997) ou é usada em outras abordagens baseadas em modelos Seq2Seq (Sutskever et al. 2014), que atuam como decodificadores. Desta forma, é possível modelar atributos espaço-temporais e produzir saídas com tamanhos variáveis, que são as denominadas características intrínsecas do problemas de tradução.

Um dos desafios neste tipo de trabalho é que os sinais consistem de três partes principais: atributos manuais envolvendo gestos feitos com as mãos, atributos não manuais, tais como expressões faciais ou postura corporal, que podem fazer parte de um sinal ou modificar o significado de um sinal, e um alfabeto manual, na qual palavras são escritas na linguagem verbal local. Naturalmente, essa é uma simplificação excessiva, a língua de

sinais é tão complexa quanto qualquer língua falada e cada língua de sinais possui dezenas de milhares de sinais, diferindo por pequenas alterações da mão, forma, movimento, posição, recurso ou contexto não manual.

O processo de reconhecimento de sinais em língua de sinais pode ser categorizado nos seguintes estágios: aquisição de dados, pré-processamento, segmentação, extração de características e classificação (Cheok et al. 2017). Contudo, no contexto do aprendizado profundo, esse processo é simplificado e automatizado, dado que as redes convolucionais (CNNs) englobam os estágios de segmentação, extração de características e classificação. A literatura apresenta diversas técnicas para a classificação de sinais, geralmente agrupadas em classificadores lineares, redes neurais e classificadores probabilísticos.

Na literatura, existem vários trabalhos que estão abordando a tradução automática a partir de texto ou áudio em linguagens faladas em animações (ou vídeos) em língua de sinais (de Araújo et al. 2012, Araujo et al. 2014, Huenerfauth 2004, Huenerfauth 2008, López-Ludeña, Morcillo, López, Ferreiro, Ferreiros & Hernandez 2014, López-Ludeña, Morcillo, López, Barra-Chicote, Cordoba & Hernandez 2014, Morrissey & Way 2013, Shoaib et al. 2013). Alguns outros trabalhos envolvem o reconhecimento de conteúdos em linguagens de sinais (por exemplo, vídeos ou imagens) em linguagens orais (Cheok et al. 2017, Akmeliawati et al. 2007, Binh & Ejima 2005, Cooper et al. 2011, Pigou et al. 2015, Masood et al. 2018). No entanto, algumas dessas soluções geralmente requerem alguns sensores ou hardware adicionais (por exemplo, luvas, braçadeiras, entre outros), o que dificulta o uso em um cenário real (Jani et al. 2018, Bessa Carneiro et al. 2016, Wu et al. 2016, Kaya et al. 2018, Kau et al. 2015, Galicia et al. 2015, Chuan et al. 2014).

Abordagens multimodais ou multifluxos surgem como uma abordagem natural para problemas que podem ser decompostos em subproblemas (Kelly et al. 2009, Kumar, Gauba, Pratim Roy & Prosad Dogra 2017, Jiang et al. 2021). Por exemplo, no contexto do reconhecimento, classificação ou tradução de Libras, sabe-se que um sinal de Libras é caracterizado por configuração das mãos, posição da mão no corpo, pelo seu movimento, orientação, movimento corporal e pelas expressões faciais, podendo esses componentes serem abordados com técnicas especializadas.

A detecção e classificação dessas diferentes partes são abordadas em diferentes estudos e apresentam técnicas e arquiteturas distintas na literatura. Portanto, é natural a concepção de arquiteturas especializadas para um domínio específico composta de diferentes fluxos, cada uma especializado em determinado sub-problema ou atributo desse domínio (Cheng et al. 2016). Por exemplo, arquiteturas baseada em fluxos óticos (do inglês, *optical flow*) podem ser empregadas para atacar o fonema de movimento, já que este modela uma representação do movimento aparente entre frames consecutivos (Simonyan & Zisserman 2014, Lim et al. 2016, Kishore et al. 2016), ou mesmo arquiteturas especializadas em detecção de postura corporal como o OpenPose (Cao et al. 2018) ou MediaPipe (Lugaresi et al. 2019), usada para lidar fonemas de expressões não manuais (Goyal et al. 2013, Ko et al. 2019, Moryossef et al. 2020).

Esse tipo de abordagem sofre com seu alto custo computacional, uma vez que os modelos são grandes e complexos. Tornando, na maioria das vezes, inviável a aplicação prática desses modelos. Uma outra limitação, no ponto de vista da otimização, é que esses modelos não modelam ou otimizam o problema de forma global, tornando o processo de

otimização (treinamento da rede neural) mais difícil e ineficiente.

No contexto brasileiro, pesquisas envolvendo o reconhecimento contínuo de Libras não são inéditas, porém, tais pesquisas acabam sendo limitadas pela ausência de conjunto de dados abertos e suficientemente robustos para o contexto de aprendizado profundo. Yauri Vidalón & De Martino (2015) propôs um sistema automático de reconhecimento de sinais baseado em *keypoints* usando Modelos Ocultos de Markov (HMMs). Já Souza et al. (2021) apresentou uma abordagem automática de reconhecimento de sinais para auxiliar a comunicação entre pacientes surdos, falantes da Língua Brasileira de Sinais (Libras) e médicos. O reconhecimento é realizado por uma cascata de duas redes neurais. A primeira, uma rede neural convolucional, codifica a entrada visual e extrai informações relevantes. A segunda, uma rede neural recorrente, aprende a mapeamento dos *features* extraídos para palavras do português brasileiro. Em relação à avaliação, o método utilizado é a avaliação automática, o sistema proposto atingiu uma acurácia de 91%. Apesar da acurácia ser boa, o experimento foi realizado com um conjunto de dados contendo apenas 280 vídeos, limitando a relevância estatística do experimento.

Adaloglou et al. (2020) apresentaram uma solução de tradução contínua de Língua de Sinais Grega para três domínios: educação, saúde e segurança pública. Para isso foi introduzido um novo corpus com 10,295 sentenças e 331 sentenças únicas. A metodologia aplicada foi baseada na rede I3D introduzida por Carreira & Zisserman (2017) e redes recorrentes bidirecionais (Huang et al. 2015). A solução atingiu um WER de 60,68% no conjunto de testes. O trabalho demonstra a viabilidade da transferência de aprendizado de uma arquitetura de vídeo (I3D) treinada para propósito geral aplicada no contexto de LS.

No trabalho de (Camgoz et al. 2018*a*), foi apresentado uma nova abordagem para a tradução automática de Língua de Sinais usando metodologias recentes em tradução automática, em especial baseado nas arquiteturas Seq2Seq2 (Sutskever et al. 2014) e com o uso de mecanismos de atenção (Bahdanau et al. 2015, Luong et al. 2015). Os experimentos computacionais foram realizados sobre conjunto de dados PHOENIX14T (Koller et al. 2015), constituído de 1066 sinais diferentes, com um vocabulário de 1066 glosas e 2887 palavras diferentes na Língua de Sinais Alemã. A solução proposta foi avaliada usando a métrica BLEU e atingiu um BLEU-4 de 19,26 no conjunto de testes.

Outro trabalho nessa linha, é o de Ko et al. (2019), que utilizou uma arquitetura Seq2Seq (Sutskever et al. 2014) baseada em redes GRU (Chung et al. 2014) para tradução da Língua de Sinais Coreana, sendo também introduzido um novo conjunto de dados denominado KETI, contendo 14,672 vídeos e 105 sentenças distintas. Um diferencial desse trabalho é o uso de informação postural provida pelo OpenPose (Cao et al. 2018) aliado ao uso de diferentes mecanismos de atenção (Bahdanau et al. 2015, Luong et al. 2015). O modelo proposto alcançou uma acurácia de tradução de 55,28% para o conjunto de validação para frases que podem ser usadas em situações de emergência.

No trabalho de Rodriguez et al. (2020), um modelo Seq2Seq foi usado para o problema de tradução da Língua Sinais de Colombiana (LSC), sua arquitetura é baseada em redes recorrentes do tipo LSTM (Hochreiter & Schmidhuber 1997). Essa abordagem tem como diferencial o uso de um codificador especial, que inclui a modelagem de movimento através do uso de fluxo ótico e dois mecanismos de *seflf-attention* cinemáticos para depen-

dências temporais curtas e longas. Os experimentos computacionais foram realizados em um novo conjunto de dados, contendo 39 sentenças gesticuladas por 13 intérpretes distintos, totalizando 1020 sentenças. Para avaliação do trabalho, foi usado a métrica BLEU-4, atingindo um BLEU máximo de 35,81 para o conjunto de testes.

Apesar de muitos trabalhos promissores, as arquiteturas de CSRL são altamente dependentes de redes recorrentes e, estas, sofrem com o problema crônico do desaparecimento (ou dissipação) do gradiente. Esse problema foi abordado através da introdução de mecanismos de atenção (Bahdanau et al. 2015, Luong et al. 2015), que evoluiu para as arquiteturas de Redes Neurais *Transformers* (RNT) (Vaswani et al. 2017), que rapidamente se tornou o estado-da-arte para diversos problemas de processamento de linguagem natural (PLN), através das arquiteturas BERT (Devlin et al. 2018) e GPT-3 (Brown et al. 2020). Posteriormente, o conceito de RNT foi expandido para problemas de Visão Computacional, através de introdução das redes *Vision Transformers* (ViT) (Dosovitskiy et al. 2020). Novamente, houve um enorme salto na literatura com a arquitetura ViT apresentando resultados superiores a arquiteturas baseadas em redes convolucionais (Trockman & Kolter 2022, Ranftl et al. 2021, Yuan et al. 2021)

Mais recentemente, a arquitetura ViT foi expandida para classificação de vídeos através da arquitetura ViViT introduzida por Arnab et al. (2021). Essa arquitetura usa dois *transformers* distintos, um para atributos espaciais e outro para classificação de atributos temporais, conseguindo assim modelar um atributo espaço-temporal. Apesar de ter sido recém introduzida, a mesma já apresenta resultados competitivos com arquiteturas clássicas, sendo assim natural o surgimento de pesquisas de tradução automática para língua de sinais baseadas em redes do tipo RNT.

No trabalho de De Coster et al. (2020), um modelo baseado em RNT foi usado para reconhecimento da Língua de Sinais Flamenga, foi aplicado uma combinação de *features* gerados por uma rede ResNet-34 (He et al. 2015) pré-treinada na ImageNet e *keypoints* gerados pelo OpenPose (Cao et al. 2018). A rede principal é uma rede *Transformer* com duas cabeças, usando uma dimensão de codificação de 512. Os experimentos computacionais foram realizados no conjunto de dados (Herreweghe et al. 2015), sendo composto de 100 classes e 104 glosas, executados por 67 intérpretes nativos. A metodologia proposta obteve uma acurácia de 74,4% para um conjunto de testes.

Camgöz et al. (2020*b*) propuseram uma arquitetura para reconhecimento e tradução de LS. A arquitetura aplicada baseia-se na rede *Transformer* e em Classificação Temporal Conexionista (CTC) (Graves et al. 2006) para vincular os problemas de reconhecimento e tradução em uma única arquitetura unificada. Os testes computacionais foram realizados sobre o corpus PHOENIX14T (Koller et al. 2015), sendo constituído de 1066 sinais diferentes, com um vocabulário de 1066 glosas e 2887 palavras diferentes na Língua de Sinais Alemã. A solução proposta atingiu um WER de 24,84, resultado este 2 vezes superior ao estado da arte anterior. A principal contribuição deste trabalho, além de ser o primeiro trabalho a aplicar a arquitetura RNT para o contexto de LS, é o fato da arquitetura ser unificada, servindo tanto para reconhecimento, quanto para tradução de LS.

A pesquisa de Camgoz et al. (2020*a*) propôs uma arquitetura multimodal baseada na arquitetura RNT para tradução de LS. A rede ao todo conta com 3 canais, dois canais baseados em imagens, um focado na mão e outro no rosto do intérprete e um terceiro canal

baseado em *keypoints*, mapeando expressões manuais e não manuais usadas na formação dos sinais. O conjunto de dados utilizado foi o PHOENIX14T (Koller et al. 2015), constituído de 1066 sinais diferentes, com um vocabulário de 1066 glosas e 2887 palavras diferentes na Língua de Sinais Alemã. O destaque deste trabalho é caracterizado pela capacidade de modelar relações inter e intra-contextuais entre diferentes elementos formadores de sinais (fonemas) na própria rede redes neural *Transformer* ao mesmo tempo em que mantém informações específicas do canal.

Yin (2020) introduziu uma arquitetura baseada em RNT que consegue ao mesmo tempo traduzir vídeos para texto e glossa para texto. Novamente, o conjunto de dados usado foi o PHOENIX14T (Koller et al. 2015), constituído de 1066 sinais diferentes, com um vocabulário de 1066 glosas e 2887 palavras diferentes na Língua de Sinais Alemã. A rede usada neste trabalho foi a GPT-2 (Radford et al. 2019*b*). Um outro resultado importante deste trabalho foi que a tradução de vídeo para texto teve resultados superiores à tradução de glosa para texto, contrariando o consenso que a tradução de glosa para texto representava um limite superior para tradução de vídeo para texto.

Bianco et al. (2022) introduziu um novo conjunto de dados para a Língua de Sinais Argentina (LSA), contendo 14.880 vídeos de sentenças extraídas do canal CN Sordos no YouTube, acompanhados de anotações de rótulos e pontos-chave para cada sinalizador. A arquitetura aplicada baseia-se na rede *Transformer*, utilizando *keypoints* como entrada. Para a avaliação do trabalho, foi utilizada a métrica WER, alcançando um BLEU máximo de 93,92 no conjunto de testes.

Tabela 3.1: Trabalhos selecionados e suas métricas.

Métodos	Ano	Arquitetura	Resultados		
			Acc	WER	BLEU
(Yauri Vidalón & De Martino 2015)	2015	HMMs	-	-	-
(Camgoz et al. 2018*a*)	2018	Seq2Seq	-	-	19,26
(Ko et al. 2019)	2019	Seq2Seq	55,28%	-	-
(Rodriguez et al. 2020)	2020	Seq2Seq	-	-	35,81
(Yin 2020)	2020	RNT	-	21,0	22,23
(Camgoz et al. 2020*a*)	2020	RNT	-	-	20,69
(Camgöz et al. 2020*b*)	2020	RNT	-	24,84	
(De Coster et al. 2020)	2020	RNT	74,7%	-	-
(Adaloglou et al. 2020)	2020	I3D+BLSTM	-	60,68	-
(Souza et al. 2021)	2021	CNN + RNN	91,00%	-	-
Bianco et al. (2022)	2022	RNT	-	93,92	-

Na Tabela 3.1 estão sintetizados os principais trabalhos relacionados ao tema deste trabalho. De maneira geral, as pesquisas brasileiras são concentradas no problema de reconhecimento isolado de sinais e dependentes de hardware específico (Yauri Vidalón & De Martino 2016, Abreu et al. 2016, Kumar, Gauba, Roy & Dogra 2017, Cerna et al. 2021), sendo a área de tradução contínua mais carente de pesquisas e a falta de base de dados um dos maiores limitadores para evolução das pesquisas da área no âmbito

nacional.

3.2 Considerações

Neste capítulo, foram apresentados alguns estudos e metodologias que podem ser aplicados ao problema de tradução automática de Libras. Até o momento desta revisão da literatura, não foram encontrados trabalhos ou soluções que apresentem uma combinação dessas características: centrado no contexto da saúde e da Libras, usando a arquitetura Seq2Seq baseada em redes *transformers* e mecanismos de atenção visual, que aborda diferentes fonemas de Libras, de forma especializada, com potencial para ser aplicado na prática clínica e com experimentos computacionais realizados sobre uma base de dados com volumes de dados representativos.

Capítulo 4

Formalização do Sistema de Tradução

A Libras é uma linguagem visual-espacial, ou seja, utiliza a visão e o espaço para compreender e produzir os sinais que formam as palavras. Em nosso trabalho, assumimos, portanto, que o sistema de tradução automática de Libras para português, tem como entrada um vídeo (sequência de imagens), contendo uma sequências de sinalizações em Libras executadas por um sinalizador A saída desejada é um conjunto de *tokens*, que podem representar tanto palavras da língua portuguesa quanto glosas, ou seja, tradução simplificada de morfemas de Libras para a língua portuguesa. Uma representação geral, em alto nível, de tal sistema é apresentada na Figura 4.1.

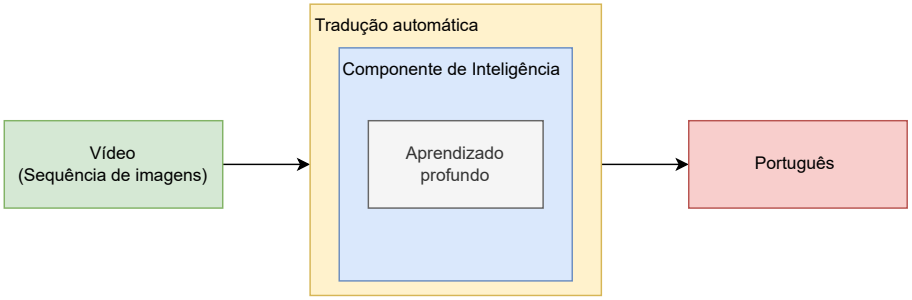

Figura 4.1: Tradução automática. Autor

Assim, neste Capítulo, serão abordados os conceitos e as definições envolvidos com a tradução automática de Libras, usados neste trabalho, necessários para compreender melhor o sistema proposto.

4.1 Tradução Automática de Libras

Tradução automática (ou tradução por computador) é o processo de tradução realizado sem a intervenção humana entre uma língua fonte e uma língua alvo, usando exclusivamente métodos automatizados assistidos por computador. No contexto do trabalho, língua fonte é a Língua de Sinais Brasileira (Libras), ou seja, é a língua que o sistema computacional *vê* para, a partir dela, fazer a tradução para outra língua (a língua alvo), a língua Portuguesa.

4.1.1 Inteligência

A parte principal do sistema de tradução automática é seu componente de *inteligência*. Neste trabalho, como indicado anteriormente, o componente de inteligência adotado é baseado na arquitetura sequência-para-sequência (Seq2Seq). Basicamente, este componente usa um modelo de aprendizado profundo especializado em mapear uma sequência fonte (sequência de imagens) em uma sequência alvo (sequência de *tokens*), fazendo assim a tradução entre as sequências. Formalmente, a arquitetura Seq2Seq produz um modelo probabilístico para tradução, dado por:

$$P(y/x) = \prod_{t}^{T'} p(\hat{y}_t = y_t|X) \tag{4.1}$$

onde cada distribuição $\mathbf{p}(\hat{\mathbf{y}}_\mathbf{t} = \mathbf{y}_\mathbf{t}|\mathbf{X})$ representada com um *softmax* sobre todos os *tokens* do vocabulário. O modelo probabilístico de tradução sabe como um *token* **y** pode ser traduzido dado uma imagem **X**, contendo uma sinalização de Libras. Esse modelo é construído durante uma etapa de otimização. O processo de otimização ou treinamento de uma rede Seq2Seq é feito através da minimização do logaritmo da probabilidade de uma tradução correta **y** dado uma sequência de origem **x**:

$$loss(x,y) = -log \cdot P(y|x) = -\sum_{i=1}^{T'} logp(\hat{z}_t = z_t|x) \tag{4.2}$$

$$\mathcal{L} = \frac{1}{\mathcal{D}} \sum_{x,y \in \mathcal{D}} loss(x,y) \tag{4.3}$$

onde $\mathcal{D}$ é o conjunto (*batch*) de treinamento. O conjunto de treinamento consiste de uma lista de pares $(\mathbf{x}, \mathbf{z})$, onde $\mathbf{x} = \mathbf{x_1}, \ldots, \mathbf{x_T}$ é a sequência origem, de comprimento **T**; e $\mathbf{y} = \mathbf{y_1}, \ldots, \mathbf{y_T}$ é a sequência destino, de comprimento **T**. $\mathbf{V_x}$ e $\mathbf{V_y}$ são os vocabulários de origem e destino, e $|\mathbf{V_x}|$ e $|\mathbf{V_y}|$ seus respectivos tamanhos. $\mathbf{E_x} \in \mathbb{R}^{|\mathbf{V_x}| \times \mathbf{m}}$ e $\mathbf{E_y} \in \mathbb{R}^{|\mathbf{V_y}| \times \mathbf{m}}$ são as matrizes de *embeddings* das sequências origem e destino. Essas matrizes são inicializados aleatoriamente e treinadas em conjunto com os demais parâmetros do modelo.

Nesse contexto, os vocabulários de origem e destino são o conjunto de *tokens* únicos que podem ser encontrados no conjunto de dados de treinamento. Mais precisamente, os *tokens* representam as palavras ou símbolos usadas para compor as diferentes sentenças do conjunto de dados. No processo de inferência, ou decodificação, as traduções são produzidas a partir da tradução mais provável dada por:

$$\hat{y} = arg\max_{y} p(y|x) \tag{4.4}$$

ou seja, dado uma sequência de sinalizações em Libras **x**, qual é a melhor sentença (mais provável) **y** em língua Portuguesa. Essa decodificação pode ter uma natureza gulosa, onde apenas a sentença de maior probabilidade é retornada, ou pode ter uma natureza heurística

usando o métodos de busca.

4.1.2 Vídeo

Um vídeo, neste contexto, é uma sequência de imagens em movimento que captura a execução de sinais de Libras por um intérprete ou um sinalizador. Os vídeos são utilizados como meio de entrada para os modelos de tradução automática, fornecendo dados visuais que representam a linguagem gestual em um formato que pode ser analisado por algoritmos de processamento de imagem e aprendizado de máquina. No âmbito da tradução automática, o vídeo deve ser processado em tempo real ou quase em tempo real, permitindo a identificação contínua dos gestos e sua conversão em línguas orais ou escritas. Para garantir a eficácia do sistema, é fundamental que os vídeos sejam de alta qualidade, apresentando boa iluminação e ângulos adequados que permitam a captura clara dos sinais executados.

4.1.3 Glosa

A glosa é um termo empregado no contexto da linguística e da tradução de línguas de sinais que se refere à representação simbólica ou anotação dos sinais, geralmente em uma língua oral ou escrita, como o português (de Albuquerque 2021). A glosa tem o objetivo de representar de forma simplificada os sinais, utilizando palavras-chave ou rótulos para descrever os elementos visuais e gestuais do sinal, incluindo sua estrutura gramatical. Cabe destacar que a glosa não é uma tradução direta nem uma transcrição detalhada dos sinais, mas sim uma convenção que descreve de maneira aproximada os significados e funções dos sinais, sem captar completamente a complexidade visual, espacial e prosódica da língua de sinais em questão.

Tabela 4.1: Exemplos de textos em português e suas respectivas representações em glosa.

Português	**Glosa**
Estou com muita dor.	EU DOR(+) [PONTO]
Estou com muita dor de barriga.	EU DOR&BARRIGA(+) [PONTO]
Preciso de ajuda urgente.	PRECISAR 2S_AJUDAR_1S URGENTE [PONTO]
Meus desmaios têm se tornado regulares.	MEU DESMAIAR TER FURACÃO REGULAR [PONTO]
Minha língua está inchada e vermelha.	MEU LÍNGUA&ÓRGÃO INCHAR VERMELHO [PONTO]

A Tabela 4.1 exemplifica como textos em português são representados por suas respectivas glosas, evidenciando a estrutura gramatical da Libras. A glosa captura aspectos importantes da gramática dessa língua de sinais, como o uso de verbos direcionais, que expressam a relação entre os sujeitos e os objetos da ação. Por exemplo, na expressão “PRECISAR 2S_AJUDAR_1S”, onde “2S"e “1S” referem-se aos pronomes “você” e “eu”, respectivamente. Além disso, a glosa demonstra o uso de intensificadores, como

em "EU DOR(+)". Neste caso, o símbolo “(+)” indica uma intensificação da dor, reforçando a gravidade da situação. A desambiguação de contexto também é uma característica marcante, como ilustrado em “DOR&BARRIGA”, onde a combinação de sinais permite distinguir entre a dor em geral e a dor específica de barriga, proporcionando clareza na comunicação. Essa análise evidencia a riqueza e a complexidade da gramática da Libras, refletindo a necessidade de uma tradução que respeite suas particularidades e nuances.

4.1.4 Corpus Paralelo

Um corpus paralelo é um conjunto de dados que contém duas ou mais línguas ou formas de representação linguística, permitindo a análise e a comparação de textos em diferentes idiomas ou sistemas de linguagem. No contexto da Língua Brasileira de Sinais (Libras), um corpus paralelo de vídeo, português e glosa se refere a um conjunto de vídeos que apresentam sinais em Libras acompanhados de suas transcrições em português e em forma de glosa.

- **Vídeos**: Os vídeos devem capturar a execução de sinais em Libras por um intérprete ou sinalizador, representando sentenças ou frases de forma clara e visível. A qualidade do vídeo deve ser suficiente para permitir a identificação dos gestos, expressões faciais e outros elementos paralinguísticos que são cruciais para a interpretação da língua de sinais.
- **Transcrição em Português**: Para cada vídeo, há uma transcrição correspondente em português, que representa o conteúdo verbal que está sendo comunicado por meio dos sinais. Esta transcrição deve ser precisa e contextualizada, garantindo que o significado original da mensagem em Libras seja mantido.
- **Glosa**: Além da transcrição em português, o corpus inclui a glosa dos sinais, que fornece uma representação detalhada dos gestos e da estrutura gramatical de Libras. A glosa é feita utilizando palavras em português, organizadas de maneira que capturem a ordem dos sinais, expressões faciais e outros elementos que fazem parte da comunicação em Libras.

4.2 Etapas no Processo de Tradução Automática

A tradução automática de Libras para uma língua oral, como o português, ou glosa, envolve uma série de etapas complexas e interdependentes. Cada uma dessas etapas desempenha um papel crucial para garantir que o sistema seja capaz de reconhecer, interpretar e traduzir os sinais de maneira precisa e contextualizada. Essa tarefa envolve as seguintes etapas: 1) aquisição de dados visuais; 2) pré-processamento de imagens; 3) extração de características; 4) reconhecimento dos sinais; 5) glosa e tradução intermediária; 6) tradução para português.

4.2.1 Aquisição de Dados Visuais

A primeira etapa consiste na aquisição dos dados visuais, que são obtidos a partir de vídeos ou sequências de imagens digitais. Esses vídeos devem capturar de forma clara e contínua os sinais em Libras realizados por um intérprete ou usuário. A qualidade dos vídeos é um fator crítico nesta fase, pois impacta diretamente a precisão do reconhecimento dos gestos. Os vídeos devem ser capturados em boas condições de iluminação e com um fundo que permita destacar os movimentos do sinalizador.

A etapa de aquisição de dados não deve ser vista apenas como um processo de captura passiva de vídeo, mas como uma etapa estratégica que visa maximizar a qualidade das informações visuais. A precisão e a robustez das etapas subsequentes, como o pré-processamento dos dados e o reconhecimento dos gestos, dependem diretamente da clareza, continuidade e detalhamento das imagens capturadas. Portanto, uma captura inadequada de dados visuais compromete todo o pipeline do sistema de reconhecimento, gerando impactos na acurácia dos resultados finais.

4.2.2 Pré-processamento de Imagens

Na fase de pré-processamento de imagens, os vídeos são preparados para o reconhecimento dos sinais. Essa etapa envolve operações de filtragem e normalização, que visam melhorar a qualidade das imagens, reduzir o ruído e isolar os gestos do intérprete do restante do cenário. Técnicas como conversão para escala de cinza, suavização, segmentação e normalização de iluminação são aplicadas para garantir que os sinais sejam adequadamente destacados, o que facilita o trabalho das etapas posteriores.

A segmentação é uma das operações mais críticas dentro do pré-processamento, pois ela isola os gestos do sinalizador, separando-os do fundo ou de outras partes do corpo que não são essenciais para o reconhecimento. Diversas técnicas podem ser empregadas para segmentar as mãos e os braços do sinalizador, como algoritmos baseados em cor da pele, análise de movimento ou métodos mais sofisticados, como segmentação por aprendizado profundo.

4.2.3 Extração de Características

A extração de características é uma das etapas mais importantes do processo. Aqui, são identificadas e extraídas informações visuais relevantes, como a posição das mãos, a orientação dos dedos, os movimentos corporais e as expressões faciais do sinalizador. Técnicas de Visão Computacional e Redes Neurais Convolucionais (CNNs) são frequentemente utilizadas para identificar padrões visuais e capturar as características relevantes dos gestos em cada quadro do vídeo. A extração de características permite que os modelos entendam a estrutura dos sinais e a linguagem corporal.

O processo de extração de características transforma uma imagem ou sequência de imagens em uma representação tensorial ou embedding, que pode ser utilizada em modelos de linguagem e em etapas subsequentes do processo de tradução. Essa representação permite que os modelos compreendam não apenas a estrutura dos sinais, mas também a informação gesto-visual que caracteriza a comunicação em Libras. Por exemplo, a

identificação precisa da posição das mãos e dos movimentos corporais pode influenciar significativamente a interpretação e a tradução dos sinais, garantindo que o significado original seja preservado.

Em suma, a extração de características forma a base sobre a qual a tradução automática de Libras é construída. Ela permite que as máquinas "vejam"e "entendam"os gestos de forma semelhante ao que um humano faria, respeitando as especificidades da comunicação em Libras.

4.2.4 Reconhecimento de Sinais

Após a extração de características, o próximo passo é o reconhecimento dos sinais. Nesta fase, um modelo de aprendizado de máquina, geralmente baseado em redes neurais profundas, classifica as características extraídas em sinais específicos da Língua de Sinais. Essa etapa envolve o treinamento dos modelos com grandes volumes de dados rotulados, ou seja, vídeos de Libras e suas transcrições, permitindo que o sistema aprenda a diferenciar os sinais de maneira precisa. O reconhecimento não se limita apenas ao movimento das mãos, mas inclui também expressões faciais e movimentos corporais, que são elementos fundamentais em Libras.

4.2.5 Glosa e Tradução Intermediária

Uma vez que os sinais foram reconhecidos, o próximo passo é a glosa. A glosa envolve a conversão dos sinais reconhecidos em uma forma escrita intermediária, utilizando palavras em português que representam os gestos feitos em Libras. Essa etapa não é a tradução final, mas uma representação textual dos sinais que mantém a ordem e estrutura da língua de sinais. A glosa serve como uma base para a tradução final e ajuda a preservar os aspectos gramaticais da língua de sinais que precisam ser levados em consideração.

4.2.6 Tradução para Português

Com base na glosa, o sistema então realiza a tradução para o português, convertendo a sequência de palavras da glosa em uma frase fluida e correta em português. Essa etapa envolve ajustar a gramática e a sintaxe para a estrutura da língua portuguesa, levando em consideração as diferenças semânticas e contextuais. O objetivo é garantir que o sentido original seja mantido e que a mensagem seja compreensível em português.

4.2.7 Refinamentos usando LLMs

A tradução automática de Libras para o Português passa por uma etapa intermediária de conversão dos sinais gestuais em uma forma escrita conhecida como glosa. A glosa preserva a estrutura gramatical de Libras, porém, para que a tradução atinja uma forma fluente em Português, é necessário converter essa glosa em uma tradução final que respeite as regras e a fluidez da língua portuguesa. Recentemente, o uso de Modelos de

Linguagem de Grande Escala (LLMs, do inglês *Large Language Models*) tem se mostrado uma abordagem promissora para realizar essa tarefa, devido à sua capacidade de gerar e transformar textos de maneira contextualizada, fluente e coerente.

Os LLMs, como o GPT-3 e o LLaMA (*Large Language Model Meta AI*), são modelos baseados em arquiteturas de *Transformers* treinados em vastas quantidades de dados textuais, o que lhes permite captar padrões semânticos, sintáticos e pragmáticos das línguas naturais. Esses modelos são capazes de realizar diversas tarefas de processamento de linguagem natural, incluindo tradução, sumarização e geração de texto, o que os torna ferramentas adequadas para a conversão de glosas em sentenças corretas e gramaticalmente precisas no Português.

A etapa de conversão de glosa em uma tradução final apresenta desafios específicos devido às diferenças entre as estruturas gramaticais de Libras e do Português. Libras, assim como outras línguas de sinais, possui uma gramática visual-espacial distinta, com ordem sintática e uso de pronomes, verbos direcionais e intensificadores que podem diferir significativamente da estrutura verbal e escrita do Português. A glosa, por sua vez, é uma representação textual direta dos sinais de Libras, mantendo essa estrutura original e, frequentemente, não sendo suficiente para gerar uma tradução fluente ou de fácil compreensão.

Os LLMs, ao serem aplicados nessa tarefa, podem ajudar a "refinar" essa glosa, transformando-a em uma sentença mais fluente em Português. Por exemplo, um modelo LLM pode receber a glosa "EU DOR&BARRIGA(+)", que corresponde à sentença em Libras "Estou com muita dor de barriga", e gerar diretamente a tradução final em Português de forma coerente e fluente. Isso é possível porque os LLMs conseguem capturar o significado dos elementos da glosa e reestruturar a frase conforme as regras da língua portuguesa.

O uso de LLMs para essa tarefa oferece várias vantagens. Primeiramente, eles são capazes de lidar com as ambiguidades presentes na glosa, como a fusão de conceitos ou a presença de múltiplos significados para um mesmo sinal. Em segundo lugar, esses modelos podem melhorar a naturalidade das traduções, eliminando a necessidade de intervenção humana para ajustar a fluidez do texto. Por fim, sua capacidade de realizar aprendizado contextual permite que as traduções resultantes sejam adaptáveis a diferentes domínios, como o médico, jurídico ou educacional, onde os sinais podem ter conotações específicas.

Um exemplo relevante no contexto dos LLMs é o LLaMA, desenvolvido pela Meta AI. O LLaMA foi treinado em uma grande quantidade de dados textuais multilíngues e é capaz de executar diversas tarefas de PLN com alto grau de precisão, incluindo tradução de idiomas e geração de texto. Sua arquitetura, baseada em *Transformers*, o torna particularmente eficaz em tarefas que exigem uma compreensão profunda de contexto e a produção de respostas coerentes em uma linguagem natural.

Quando aplicado à tarefa de conversão de glosa em Português, o LLaMA pode ser ajustado para reconhecer as nuances da glosa de Libras e realizar a reestruturação necessária para que a frase resultante esteja de acordo com as convenções gramaticais do Português. Por exemplo, o LLaMA pode lidar com verbos direcionais típicos de Libras, como "2S_AJUDAR_1S", onde "2S"e "1S" se referem aos pronomes "você" e "eu", res-

pectivamente, e reestruturar essa informação para gerar uma frase fluida em Português, como “Você pode me ajudar?”. Além disso, o modelo pode interpretar corretamente intensificadores como “(+)” e realizar a tradução semântica correta.

4.3 Considerações

Neste capítulo, foram apresentados os conceitos, definições e etapas que caracterizam o problema e a proposta de solução para a tradução automática da Língua Brasileira de Sinais (Libras) para o Português. Neste sentido, foram definidos o processo de tradução automática de Libras para o Português, bem como os insumos necessários à execução e as etapas envolvidas nesse processo.

Uma solução de tradução automática de Libras baseada em inteligência artificial e visão computacional deve integrar esses processos de maneira coesa, visando garantir a eficácia e a precisão das traduções. Para tanto, foram planejadas etapas de aquisição e projeto da base de dados, pré-processamento dos dados, definição das arquiteturas neurais de tradução automática, avaliação automática e treinamento dos modelos.

O próximo capítulo descreverá a implementação da solução, fundamentando-se no embasamento do Capítulo 2 e nas definições, conceitos e proposta de solução apresentadas neste capítulo, e detalhando como cada uma das etapas planejadas será executada.

Capítulo 5

Implementação

Neste Capítulo, descrevemos os detalhes de implementação da solução proposta no Capítulo anterior (Cap. 4) para tradução automática de Libras.

A Seção 5.1 apresenta a base de dados e o pré-processamento desenvolvido para solução proposta. A Seção 5.2 apresenta a arquitetura usada na solução proposta. A Seção 5.3 apresenta as métricas computacionais para avaliação do solução desenvolvida. A Seção 5.4 apresenta as configurações e parâmetros de treinamento do modelo.

5.1 Base de dados

Inicialmente, detalha-se o processo de aquisição e pré-processamento dos dados. Considerando a ausência de bases de dados públicas disponíveis na literatura específica sobre tradução automática de Libras, especialmente no domínio da saúde, desenvolveu-se uma base de dados própria, com o suporte de especialistas em Libras para assegurar precisão linguística. A base foi criada para atender às necessidades deste trabalho e contém 100 amostras de vídeo para cada uma das 105 sentenças selecionadas, totalizando 10.500 vídeos. Todo o conteúdo é oriundo do domínio da saúde.

5.1.1 Planejando e Criando a Base de Dados

A seleção dos intérpretes foi realizada com o objetivo de representar a variabilidade étnica brasileira, com diferentes tonalidades de pele, perfis corporais e faixas etárias. Essas amostras foram capturadas a partir de uma câmera de celular em qualidade HD (1080p) a 30 fps, em ambientes de gravação padronizados, contendo ao todo 10.500 vídeos. O ambiente de gravação foi configurado com fundo branco, o intérprete posicionado a 40 cm dele, a câmera colocada a uma altura de 1,5 metros e a uma distância de 2,5 metros do intérprete.

Cada vídeo foi gravado por um dos 10 participantes selecionados, em 10 sessões distintas, a fim de garantir uma variação amostral. O desenvolvimento da base de dados, conforme ilustrado na Fig. 5.1, inicia-se com a definição das sentenças baseadas em cenários de triagem clínica, seguida da tradução para glosa e da identificação dos sinais utilizados em cada gravação.

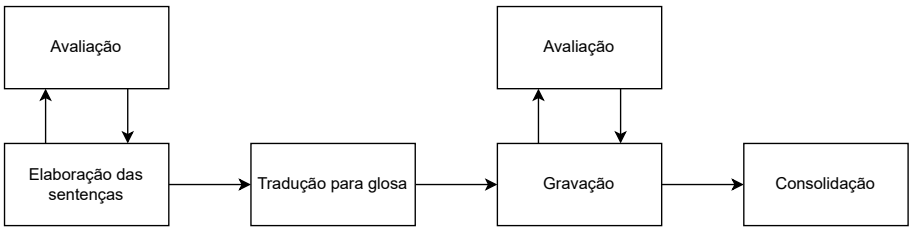

Figura 5.1: Projeto do conjunto de dados. Fonte: Autor

Um dos principais desafios enfrentados durante o desenvolvimento da base foi lidar com o fenômeno do regionalismo, característico tanto da Libras quanto de outras línguas naturais. Esse fenômeno refere-se à existência de variações regionais de sinalização para representar o mesmo conceito, seja ele abstrato ou concreto. Embora fosse ideal capturar a maior variabilidade linguística possível, tal abordagem demandaria uma quantidade considerável de recursos, além de estender o tempo necessário para a coleta e tratamento dos dados. Para mitigar essa limitação, optou-se por utilizar sinais de aceitação ampla em grande parte do território nacional. Essa decisão fundamentou-se na experiência e no conhecimento especializado dos intérpretes e profissionais de Libras envolvidos no projeto.

A partir da lista de sentenças validadas, deu-se início à etapa de gravação dos vídeos que compõem a base de dados, seguida por etapas de validação da qualidade dos vídeos produzidos até a consolidação final do corpus de dados. Após a validação da base, foi necessário preparar e transformar esses dados para que estivessem aptos a serem utilizados no processo de aprendizado de máquina supervisionado.

5.1.2 Pré-processamento de dados

O conjunto de vídeos foi transformado em um conjunto de frames, o número de frames extraídos de cada vídeo foi fixado em 40. Para cada frame, a região de interesse, retângulo ao redor do intérprete foi recortada usando o YOLO[1] (Redmon et al. 2015). Esse processo é necessário para reduzir a quantidade de informação desnecessária e ajudar o modelo durante as etapas de treinamento e inferência. Por fim, os *frames* recortados tiveram suas dimensões padronizadas em 128x128 pixels, resultando em uma dimensão de entrada para a rede de $(\mathbf{batch\_size}, \mathbf{40}, \mathbf{128}, \mathbf{128}, \mathbf{3})$. As sentenças foram então divididas em blocos de *tokens* únicos, processo conhecido como tokenização (Webster & Kit 1992), e vetorizadas para uma representação numérica inteira (Figura 5.2). Essa sequência de processos é necessária para diminuir o custo computacional e melhorar o desempenho do modelo.

Redes neurais usualmente requerem um tensor com dimensão fixa; no entanto, sentenças em Libras e, consequentemente, seus vídeos, possuem durações variadas, resultando em um número não padronizado de frames. No contexto de arquiteturas Seq2Seq, esse problema é tratado com técnicas de preenchimento (do inglês, *padding*), nas quais o tensor de entrada é padronizado pela adição de um token <*pad*>. Esse token é ignorado no

[1] https://github.com/ultralytics/yolov5

processo de treinamento por meio de uma máscara na função de perda, de modo que sua adição não interfere no aprendizado da rede. Entretanto, para reduzir o custo computacional dos experimentos, optou-se pelo uso de amostragem: vídeos com número de frames $\mathbf{T} < \mathbf{40}$ passam por superamostragem, e vídeos com número de frames $\mathbf{T} > \mathbf{40}$ passam por subamostragem.

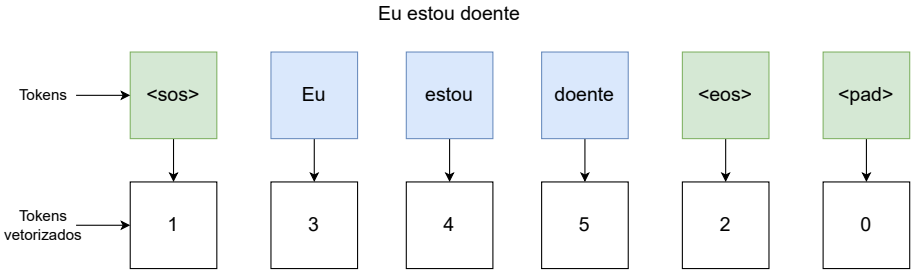

Figura 5.2: Processo de tokenização e vetorização das sentenças. Fonte: Autor

Durante o processo de tokenização, também são adicionados alguns tokens especiais: <sos>, <eos> e <pad>, que indicam, respectivamente, o início da sentença, o final da sentença e um token de preenchimento. Este último tem como objetivo garantir que todas as entradas tenham um tamanho fixo. Por exemplo, se a rede espera sentenças com 50 tokens e a sentença atual tiver 5 tokens, então serão adicionados 45 tokens <pad> à esquerda dos tokens originais da sentença.

Ao final do pré-processamento, o conjunto de treinamento consiste em uma lista de pares $(\mathbf{x}, \mathbf{y})$, onde $\mathbf{x} = \mathbf{x_1}, \ldots, \mathbf{x_T}$ é uma sequência de frames de comprimento $\mathbf{T} = 40$ e $\mathbf{y} = \mathbf{y_1}, \ldots, \mathbf{y_T}$ é uma sequência de tokens de comprimento $\mathbf{T} = \mathbf{50}$. $\mathbf{V_y}$ é o vocabulário de tokens e $|\mathbf{V_y}|$ seu respectivo tamanho.

5.1.3 Processo de aumentação de dados

Após o pré-processamento, as imagens geradas passam por um processo de aumentação de dados (no inglês, *data augmentation*). Esse processo consiste, essencialmente, na ampliação artificial do volume de dados por meio da geração de novas instâncias com base nas já existentes no conjunto de dados. A geração dessas novas instâncias ocorre por meio da aplicação de técnicas de processamento digital de imagens, nas quais ruídos artificiais são introduzidos para simular variações de iluminação, perda de foco, distorções espaciais, entre outros artefatos típicos de gravações realizadas com celulares em condições não controladas. Além disso, também são aplicadas transformações lineares com parâmetros aleatórios, como rotações, deslocamentos e variações de escala.

Esse processo é essencial para adicionar maior variabilidade durante o treinamento da rede, aumentando, assim, a resistência do modelo contra o sobreajuste. A transformação de espelhamento vertical é particularmente relevante, pois um mesmo sinal em Libras pode ser executado com a alternância entre as mãos direita e esquerda. Essa etapa assume ainda mais importância em redes do tipo *Transformers*, que possuem uma quantidade significativa de parâmetros treináveis, tornando-as especialmente suscetíveis ao sobreajuste.

Tabela 5.1: Transformações aplicadas aos dados de treinamento

Transformação	Variação ou parâmetros	Justificativa
Rotação	(-20,20)	Variabilidade
Translação	(-50,50)	Variabilidade
Recorte aleatório	(64, 64)	Variabilidade
Desfocagem	σ=(0.1, 5)	Perda de foco
Contraste	(-25,25)	Variação de luz
Intensidade	(-15,15)	Variação de luz
Espelhamento vertical	$p = 0.5$	Variabilidade

Dessa forma, o aumento de dados contribui para melhorar a generalização do modelo, tornando-o mais robusto a variações nos dados de entrada durante o processo de inferência. Isso permite que o modelo desempenhe de maneira mais eficaz ao lidar com dados não vistos, reduzindo a chance de erros causados por pequenas variações nas condições dos vídeos ou nas execuções dos sinais.

5.1.4 Divisão da base de dados

O processo de divisão da base de dados para o treinamento buscou ser o mais condizente possível com o método científico e, principalmente, resiliente ao problema de sobreajuste, garantindo, assim, que os resultados obtidos representem a real capacidade de generalização do modelo, ou seja, que não sejam camuflados pelo problema supracitado.

Portanto, para todos os experimentos computacionais que serão realizados, 70% dos vídeos que compõem a base (cerca de 7350 vídeos) serão destinados ao treinamento dos modelos de aprendizado profundo, 20% à etapa de validação e 10% à etapa final de testes. Para o processo de teste, o intérprete utilizado não deverá ter sido visto pelo modelo em nenhum momento durante o treinamento; essa escolha tem como finalidade atestar a real capacidade de generalização dos modelos.

Os dados resultantes do processo de pré-processamento estão prontos para serem usados no processo de aprendizado de máquina supervisionado (ver Fig. 5.3). Esse processo é composto de uma etapa de modelagem, onde será selecionado um algoritmo, também referido como modelo ou arquitetura, esse modelo será treinado, processo de otimização responsável por minimizar ou maximizar as métricas de avaliação, sobre um conjunto de dados rotulados, com exemplos de entrada $\mathbf{x} = \mathbf{x_1}, \ldots, \mathbf{x_T}$ e saída $\mathbf{y} = \mathbf{y_1}, \ldots, \mathbf{y_T}$, isto é, pares de vídeos de Libras (sequência de imagens) e suas respectivas traduções (*tokens*). Por fim, a etapa de predição, usa o modelo treinado para realizar traduções sobre novos dados. Portanto, o componente mais importante para esse processo de aprendizado supervisionado de tradução automática é o modelo ou arquitetura, que representa o componente de inteligência do sistema.

A próxima seção abordará a arquitetura do modelo, que desempenha um papel fundamental no processo de aprendizado supervisionado para a tradução automática de Libras.

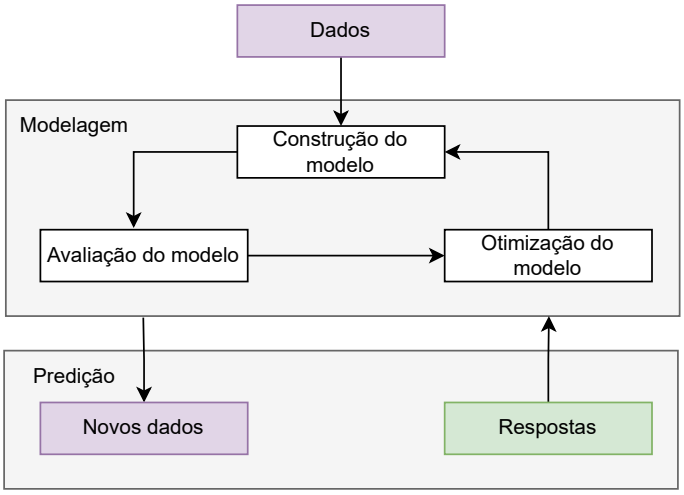

Figura 5.3: Processo de aprendizado de máquina supervisionado. Fonte: Autor

5.2 Arquiteturas

Nesta seção, serão discutidas as diferentes abordagens arquiteturais, suas características e como elas influenciam a eficácia do sistema na realização de traduções precisas e confiáveis. A escolha da arquitetura adequada é crucial, pois determina a capacidade do modelo de capturar a complexidade dos sinais em Libras e gerar traduções de qualidade a partir de novos dados.

5.2.1 Seq2seq com GRU

A primeira arquitetura descrita será uma arquitetura Seq2Seq (Sutskever et al. 2014) baseada GRU com mecanismos de atenção (Luong et al. 2015) e atenção visual (Zhang et al. 2018). Essa arquitetura será usada como referência para as demais arquiteturas. Uma visão geral da arquitetura pode ser visualizada na Figura 5.4.

Encoder

O bloco de codificador (Ver Figura 5.5) é formado por três componentes: um *backbone*[2], uma módulo de atenção visual e uma rede GRU. O *backbone* é um extractor de *features*, isto é, uma rede CNN-2D pré-treinada sobre o conjuntos de dados *ImageNet* [3], para isso, a camada final responsável gerar uma distribuição de probabilidade (classificação) é removida para se ter acesso ao mapa de *features* gerado pela última camada convolucional. Portanto, o *backbone* realiza a compressão de dimensão de cada *frame* do vídeo $\mathbf{I}(\mathbf{t})$ em um vetor 1D $\mathbf{h}(\mathbf{t})$, transformação sumarizada a seguir:

[2]A rede principal (básica) usada para extrair características da imagem de entrada
[3]https://www.image-net.org/

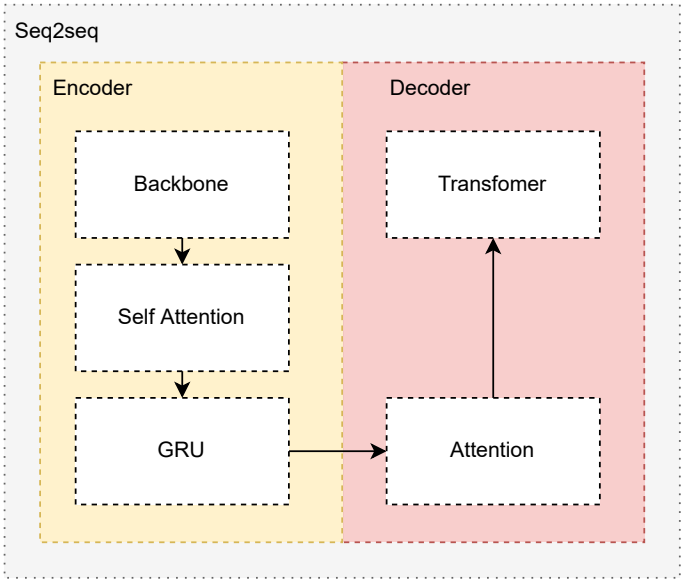

Figura 5.4: Arquitetura Seq2seq com GRU. Fonte: Autor

$$f_{CNN}(I(t)) = h(t) \tag{5.1}$$

Em seguida, esse mapa de *features* alimenta um módulo de autoatenção visual. Esse componente é responsável por aprender a ponderação do mapa de *features*, permitindo que a rede identifique quais regiões de uma imagem são mais relevantes para o contexto de uma tradução específica. Dessa forma, a arquitetura pode focar nas áreas da imagem que contêm informações cruciais, melhorando a qualidade e a precisão das traduções geradas.

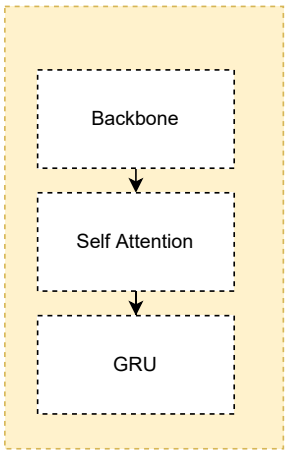

Figura 5.5: Codificador da arquitetura Seq2seq com GRU. Fonte: Autor

Por fim, a rede GRU recebe a sequência $\mathbf{h(t)}$ e produz outro vetor de contexto denominado $\mathbf{z(t)}$. A cada passo de tempo, a entrada para o codificador GRU é o *embedding* do mapa de *features* do *frame* atual $\mathbf{f_{CNN}(I(t))}$, bem como o estado oculto do passo de tempo anterior $\mathbf{h_{t-1}}$, e o codificador GRU emite um novo estado oculto $\mathbf{h_t}$. Podemos pensar no estado oculto como uma representação vetorial do vídeo até o instante de tempo **t**. Portanto, podemos representar o codificador como:

$$h_t = encoder(f_{CNN}(I(t)), h_{t-1}) \tag{5.2}$$

Uma vez que o *frame* final, tenha sido passada para o GRU através da camada de *embedding*, usamos o estado oculto final $\mathbf{h_t}$, como o vetor de contexto **z**, ou seja, $\mathbf{h_t = z}$. Esta é uma representação vetorial de todo o vídeo (sequência de *frames*). Portanto, a função do codificador é produzir uma representação vetorial do vídeo, em uma dimensão fixa e usualmente menor do que a dimensão original dos dados. Para o experimento foram estudados vetores de *embeddings*[4] com dimensões 256, 512 ou 768.

Decoder

O bloco de decodificador (Ver Figura 5.6) é formado por dois componentes: um módulo de atenção e uma rede GRU. De posse do vetor de contexto $\mathbf{h_t = z}$, podemos começar a decodificá-lo para obter a sentença destino (tradução). A cada passo de tempo **t**, a entrada para o decodificador é o *embedding* **d**, da palavra atual $\mathbf{d(y_t)}$, bem como o estado oculto do passo de tempo anterior $\mathbf{s_{t-1}}$, onde o estado oculto inicial do decodificador $\mathbf{s_0}$, é o vetor de contexto **z**, ou seja, o estado oculto inicial do decodificador é o estado oculto do codificador final $\mathbf{s_0 = z = h_t}$. Assim, semelhante ao codificador, podemos representar o decodificador como:

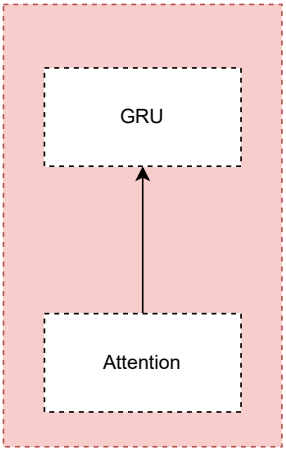

Figura 5.6: Decodificador da arquitetura Seq2seq com GRU. Fonte: Autor

$$s_t = decoder(d(y_t), s_{t-1}) \tag{5.3}$$

[4]Projeção de um vetor de alta dimensão (esparso) em um vetor de baixa dimensão.

No decodificador, precisamos ir do estado oculto **s** para uma palavra real, portanto, a cada passo de tempo **t**, usamos para prever, o que pensamos ser a próxima palavra na sequência, para isso é usada uma camada linear **f**:

$$\hat{y}_t = f(s_t) \tag{5.4}$$

O módulo de atenção tem seu funcionamento baseado no cálculo um vetor de atenção **a**, um vetor com a mesma dimensão da sequência de entrada. Esse vetor tem a propriedade que cada elemento está entre 0 e 1 e a soma de todos os elementos do vetor deve ser igual a 1. Essa camada tem como entrada o estado oculto do decodificador, $\mathbf{s_{t-1}}$, e todos os estados ocultos do codificador empilhados em ambas as direções **h** (bidirecional), resultando em um vetor de atenção $\mathbf{a_t}$.

Função objetivo

A função da rede RNN é estimar a probabilidade condicional $\mathbf{p}(\mathbf{y}|\mathbf{x})$, onde $\mathbf{x} = \mathbf{x_1}, \ldots, \mathbf{x_t}$ é a sequência de entrada (sequência de *frames*) e $\mathbf{y} = \mathbf{y_1}, \ldots, \mathbf{y_t}$ é a sequência de saída, os comprimentos da sequência de entrada $\mathbf{T'}$ e da sequência de saída T podem ser diferentes, porém, limitados por $\mathbf{T_{max}} = 40$ e $\mathbf{T'_{max}} = 50$, respectivamente, portanto, o modelo pode ser formalizado como:

$$P(y/x) = \prod_{t}^{T'} p(\hat{y}_t = y_t|X) \tag{5.5}$$

Sendo cada distribuição $\mathbf{p}(\hat{\mathbf{y}}_\mathbf{t} = \mathbf{y_t}|\mathbf{X})$ representada com um *softmax* sobre todos os *tokens* do vocabulário. O processo de otimização da rede é feito através da minimização do logaritmo da probabilidade de uma tradução correta **y** dado uma sequência de origem **x**:

$$loss(x,y) = -log \cdot P(y|x) = -\sum_{i=1}^{T'} logp(\hat{z}_t = z_t|x) \tag{5.6}$$

$$\mathcal{L} = \frac{1}{\mathcal{D}} \sum_{x,y \in \mathcal{D}} loss(x,y) \tag{5.7}$$

onde $\mathcal{D}$ é o *batch* de treinamento. No processo de inferência, as traduções são produzidas a partir da tradução mais provável dada por:

$$\hat{y} = arg\max_{y} p(y|x) \tag{5.8}$$

5.2.2 Seq2seq com Redes Neurais Transformadoras

A próxima arquitetura que foi estudada (Ver Fig.5.7) é uma arquitetura Seq2Seq baseada em Redes Neurais Transformadoras (RNT), como explicado na Seção 2, as redes transformadoras são um tipo de rede totalmente baseada em mecanismos de atenção e, atualmente, o estado da arte para vários problemas de processamento de linguagem natural.

Para que fosse possível usar essa arquitetura foi utilizada redes CNN para fazer o *embedding* do vídeo (sequência de imagens). Novamente foi utilizada uma rede ResNet-18 pré-treinada sobre o conjunto de dados *ImageNet*, onde sua camada completamente conectada foi removida para se ter acesso ao mapa de *features* gerado pela última camada convolucional. Portanto, o processo de *embedding* converte uma entrada com dimensão $(40, 128, 128, 3)$ em um tensor de dimensão $(40, d_{size})$, onde $\mathbf{d_{size}}$ é a dimensão de *embedding*, podendo assumir os valores de 256, 512 ou 768. A rede apresenta, ao todo, 6 blocos codificadores e 6 blocos decodificadores, cada um contendo 8 cabeças (blocos de atenção).

Encoder

O bloco de codificador é formado por três componentes: um *backbone*, um módulo de atenção multi-cabeça e uma rede feedforward. O *backbone* é um extractor de *features*, isto é, uma rede CNN-2D pré-treinada sobre o conjunto de dados *ImageNet*, para isso, a camada final responsável por gerar uma distribuição de probabilidade (classificação) é removida para se ter acesso ao mapa de *features* gerado pela última camada convolucional. Portanto, o *backbone* realiza a compressão de dimensão de cada *frame* do vídeo $\mathbf{I}(\mathbf{t})$ em um vetor 1D $\mathbf{h}(\mathbf{t})$, transformação sumarizada a seguir:

$$f_{CNN}(I(t)) = h(t) \tag{5.9}$$

Em seguida, esse mapa de *features* alimenta o módulo de atenção multi-cabeça. Esse componente permite que o modelo atenda conjuntamente a informações de diferentes subespaços de representação em diferentes posições. Para esse experimento, foi usado 6 codificadores empilhados (Ver Figura 5.9). Todos os codificadores têm a mesma arquitetura, sendo formados por um bloco de autoatenção e uma rede feedforward. Para o experimento, foram estudados vetores de *embeddings* com dimensões 256, 512 ou 768. As dimensões de saída são fixadas para facilitar o uso de conexões residuais entre os blocos de atenção e a rede feedforward.

Decoder

O decodificador consiste em todos os componentes acima mencionados mais dois novos. Como antes: A sequência de saída é alimentada em sua totalidade e os *embeddings* de palavras são calculados, então a codificação posicional é aplicada novamente e os vetores são passados para o primeiro bloco do decodificador.

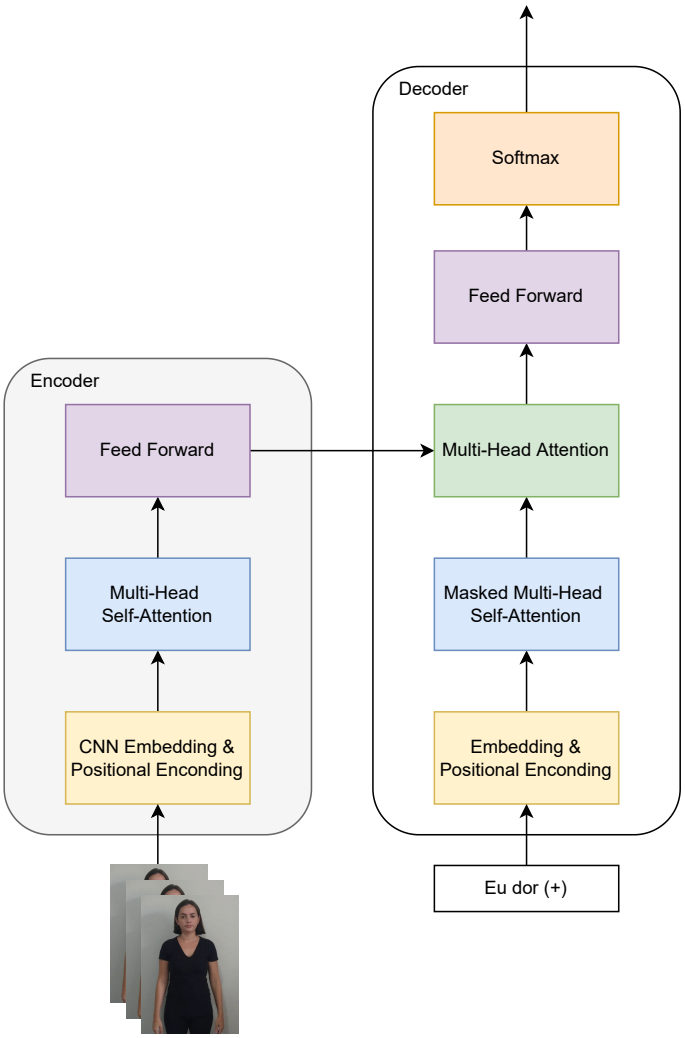

Figura 5.7: Arquitetura Seq2seq com RNT. Fonte: Autor

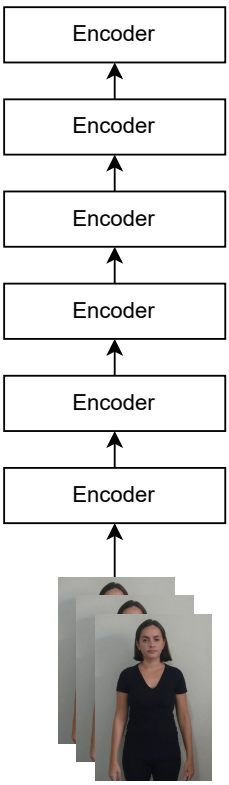

Figura 5.8: Codificador da arquitetura com RNT. Fonte: Autor

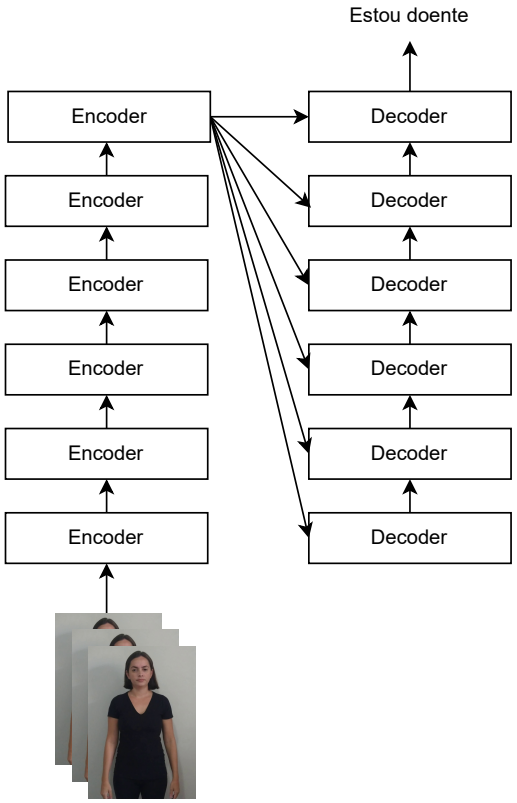

Figura 5.9: Decodificador da arquitetura com RNT. Fonte: Autor

O bloco decodificador também é formado por 6 blocos iguais em sequência. Semelhante ao codificador, são empregadas conexões residuais em torno de cada uma das subcamadas, seguidas de normalização de camada.

Além disso, a camada de auto-atenção do decodificador é modificada para evitar que, durante o processo de treinamento, onde o modelo tem acesso a toda sequência de tokens, sejam usadas informações posteriores ao tempo atual **t**, ou seja, informações do futuro. Neste trabalho, foram usadas 8 camadas de atenção (ou cabeças), sendo que cada cabeça produz saídas de dimensão $\mathbf{d_{size}/8}$. A saída final é transformada através de uma camada linear final e as probabilidades de saída são calculadas com a função *softmax*. As probabilidades de saída predizem o próximo token $\hat{y}$ na sentença de saída.

$$\hat{y} = arg\max_{y} p(y|x) \tag{5.10}$$

As redes *transformers* são altamente suscetíveis ao problema de sobreajuste, ou seja, quando o modelo aprende excessivamente os padrões específicos do conjunto de treinamento e perde capacidade de generalização. Para mitigar esse problema, aplica-se a técnica de *dropout*, que consiste em "desativar" aleatoriamente uma fração dos neurônios durante o treinamento, evitando que o modelo dependa demais de conexões específicas. Um *dropout* de 0.1 é aplicado na geração do *embedding* e um *dropout* de 0.2 é aplicado no codificador e no decodificador.

Função objetivo

A função da rede é estimar a probabilidade condicional $\mathbf{p(y|x)}$, onde $\mathbf{x} = \mathbf{x_1}, \ldots, \mathbf{x_t}$ é a sequência de entrada (sequência de *frames*) e $\mathbf{y} = \mathbf{y_1}, \ldots, \mathbf{y_t}$ é a sequência de saída. Os comprimentos da sequência de entrada $\mathbf{T}'$ e da sequência de saída T podem ser diferentes, porém, limitados por $\mathbf{T_{max}} = 40$ e $\mathbf{T'_{max}} = 50$, respectivamente, portanto, o modelo pode ser formalizado como:

$$P(y/x) = \prod_{t}^{T'} p(\hat{y}_t = y_t|X) \tag{5.11}$$

Sendo cada distribuição $\mathbf{p(\hat{y}_t = y_t|X)}$ representada com um *softmax* sobre todos os *tokens* do vocabulário. O processo de otimização da rede é feito através da minimização do logaritmo da probabilidade de uma tradução correta **y** dado uma sequência de origem **x**:

$$loss(x,y) = -log \cdot P(y|x) = -\sum_{i=1}^{T'} logp(\hat{z}_t = z_t|x) \tag{5.12}$$

$$\mathcal{L} = \frac{1}{\mathcal{D}} \sum_{x,y \in \mathcal{D}} loss(x,y) \tag{5.13}$$

onde $\mathcal{D}$ é o *batch* de treinamento.

5.3 Métricas de Avaliação

As métricas de avaliação são essenciais para medir a eficiência dos modelos dos modelos estudados. As métricas de avaliação que serão usadas são as principais métricas usadas para avaliação de tradução automática (BLEU) e para reconhecimento contínua de língua de sinais (WER).

A métrica WER (do inglês, *Word Error Rate*) é a métrica padrão para avaliação de reconhecimento contínuo de língua de sinais. Ela mensura a similaridade entre as predições e as glosas verdadeiras, através da porcentagem de palavras incorretas em função do total de palavras processadas. Mais especificamente, é calculado a quantidade operações (substituição, deleção e inserção) necessárias para transformar a sequência de *tokens* preditos $\mathbf{y}^{'} = \mathbf{y_1^{'}}, \ldots, \mathbf{y_T^{'}}$ na sequência de *tokens* verdadeiros $\mathbf{y} = \mathbf{y_1}, \ldots, \mathbf{y_T}$

$$WER(y^{'}, y) = \frac{S + D + I}{N} \tag{5.14}$$

onde **S** é o número total de substituições, **I** o número de inserções, **D** o número de deleções, e **N** é o total de palavras processadas.

BLEU

A métrica BLEU (do inglês, *Bilingual Evaluation Understudy*) (Papineni et al. 2002) é a métrica mais popular para avaliação de tradução automática. Sendo projetada para substituir e automatizar a avaliação humana em cenários onde múltiplas avaliações são necessárias.

$$BLEU = BP \cdot exp\left(\sum_{n=1}^{N} w_n log p_n\right) \tag{5.15}$$

onde $\mathbf{p_n}$ é a precisão modificada para n-grama, a base de **log** é a base natural **e**, $\mathbf{w_n}$ é o peso entre 0 e 1 para $\mathbf{log}(\mathbf{p_n})$ e $\sum_{\aleph=\nVdash}^{\mathbb{N}} \precsim_{\aleph} = \nVdash$, e BP é a penalidade de brevidade para penalizar traduções automáticas curtas.

$$BP = \begin{cases} 1 & \text{if } c > r \\ exp(1 - \frac{r}{c}) & \text{if } c \leq r \end{cases} \tag{5.16}$$

onde **c** é o número de unigramas (comprimento) em todas as sentenças candidatas, e **r** é o melhor comprimento de correspondência para cada sentença candidata no corpus. Foi usado um BLEU com $\mathbf{N = 4}$ e $\mathbf{w_n = 1/N}$.

Uma pontuação BLEU de zero significa que não há correspondência com a referência, enquanto uma pontuação de um (ou 100%) significa uma correspondência exata entre a hipótese e a referência. As correspondências exatas são raras e as pontuações BLEU

geralmente estão bem abaixo de 100%. Um grande problema com o BLEU é que ele procura apenas correspondências exatas de n-gramas com uma referência. No entanto, uma tradução pode ser perfeita sem ter nenhuma palavra em comum com uma determinada tradução de referência.

Além das métricas WER e BLEU , durante o treinamento, também foram avaliadas a acurácia e a perda do modelo. Essas métricas são utilizadas para monitorar o progresso do treinamento por meio de um mecanismo de parada antecipada. Assim, se a melhoria dessas métricas cessar por um período superior a 10 épocas, o treinamento é finalizado. Essa abordagem visa evitar o sobreajuste e garantir que o modelo alcance um desempenho ideal durante as inferências com dados não vistos durante o treinamento. Dessa forma, assegura-se que o modelo não apenas memorize as amostras do conjunto de treinamento, mas que também generalize bem para novos dados, mantendo a precisão e a confiabilidade nas traduções realizadas.

5.4 Treinamento

Para o treinamento do modelo, foi utilizado o aprendizado supervisionado, com os seguintes hiperparâmetros (Ver Tabela 5.2), *Learning rate* foi inicializada em $1e^{-4}$ e a *weight decay* em $1e^{-5}$ usando o otimizador Adam (Kingma & Ba 2014) e a biblioteca PyTorch (Paszke et al. 2019). Também foi aplicada um decaimento exponencial da taxa de aprendizado $lr = 1e^{-4}/(1+\gamma t)$, onde $\gamma = 0.995$. Os modelos são treinados com um *batch size* de 6, com um total de 50 iterações. Do total de amostras presentes no conjunto de dados, 70% (7350 instâncias) foram utilizadas para o treinamento da rede, 20% (2010 instâncias) para validação durante treinamento e 10% (1050 instâncias) para teste com modelo treinado. Vale lembrar que o intérprete do conjunto de teste é o único que não participa do treinamento, ou seja, amostras desconhecidas pelo modelo, é este conjunto que é utilizado para validar o modelo já treinado.

Tabela 5.2: Especificação dos Hiperparâmetros

Hiperparâmetro	Valor
Taxa de aprendizado	1e-4
Regularização L2	1e-3
Otimizador	AdamW
Frames por vídeo	40
Dimensão da imagem	(128,128,3)
Batch size	6
Dropout	0,5
Épocas	50

5.4.1 Fine-tuning do LLaMA

A implementação do *fine-tuning* do modelo LLaMA 3.2 para a tarefa de tradução de glosa em Português envolve uma série de etapas metódicas e rigorosas que visam adaptar o modelo a essa tarefa específica. Inicialmente, é necessário reunir um conjunto de dados representativo e diversificado, composto por glosas de Libras e suas respectivas traduções em Português. Esse corpus deve refletir diferentes contextos e temas, garantindo que o modelo aprenda a lidar com as variações linguísticas e culturais que podem surgir nas traduções.

Para isso, será utilizado o corpus de treinamento do motor de tradução do VLibras[5]. Este corpus, que é generalista e foi desenvolvido por especialistas em Libras, é composto por 120 mil sentenças e suas respectivas glosas em Libras. No contexto deste trabalho, vamos usar o corpus na direção inversa, ou seja, traduzir glosa em português. Essa abordagem permite explorar as ricas representações linguísticas presentes nas glosas, aproveitando a estrutura gramatical e semântica delas para facilitar a conversão precisa para o português. Ao empregar esse corpus, pretendemos não apenas melhorar a qualidade da tradução, mas também garantir que as nuances e especificidades da linguagem de sinais sejam adequadamente refletidas na língua portuguesa, assegurando assim uma comunicação mais efetiva e natural entre as duas formas de expressão.

Com os dados preparados, a fase de *fine-tuning* propriamente dita pode ser iniciada. Para isso, o LLaMA 3.2 [6], é carregado com seus pesos pré-treinados, que já contêm um vasto conhecimento sobre a linguagem. O modelo escolhido para o fine-tuning será o modelo de 1 bilhão de parâmetros, pois esta versão é mais leve e menos custosa para treinar. Essa característica torna o modelo mais acessível em termos de recursos computacionais, permitindo que ele seja utilizado em ambientes com limitações de hardware, além de reduzir o tempo necessário para o processo de treinamento.

Para *fine-tuning*, foi utilizado o aprendizado supervisionado, com os seguintes hiperparâmetros, *Learning rate* foi inicializada em $1e^{-4}$ usando o otimizador Adam (Kingma & Ba 2014) e a biblioteca PyTorch (Paszke et al. 2019). Os modelos são treinados com um *batch size* de 6, com um total de 1 iteração.

A Tabela 5.3 mostra as especificações de hardware da máquina em que todos os experimentos computacionais foram executados. Todos os testes foram realizados no ambiente Linux através da distro Ubuntu 18.04 LTS, com a versão 10.1 do CUDA, plataforma de computação paralela da NVIDIA e a versão 7.6 do cuDNN, biblioteca de primitivas para processamento otimizado de redes neurais convolucionais em placas de vídeo da NVIDIA.

5.4.2 Resumo da Implementação

Neste capítulo, apresentamos a arquitetura da solução proposta para a tradução automática de Libras para Português, detalhando seus componentes e a implementação subjacente. Discutimos os aspectos fundamentais que integram o sistema, incluindo a base

[5]https://www.gov.br/governodigital/pt-br/acessibilidade-e-usuario/vlibras
[6]https://www.llama.com/

Tabela 5.3: Especificações do Hardware

Especificações	CPU	GPU
Fabricante	Intel	Nvidia
Número de Cores	4	3840
Frequência de Clock (GHz)	2,40	1,19
Memória (GB)	32	16

de dados, que serve como base essencial para o treinamento e avaliação do modelo, e o pré-processamento necessário para assegurar que os dados estejam adequadamente formatados para análise.

A descrição da arquitetura detalhou as escolhas feitas em relação aos algoritmos e estruturas utilizadas, evidenciando como cada componente foi selecionado para maximizar a eficácia da tradução automática. Além disso, abordamos as métricas computacionais que possibilitam a avaliação da performance da solução desenvolvida, permitindo uma compreensão clara dos resultados obtidos e sua relevância no contexto da tradução automática.

O capítulo a seguir descreve os experimentos realizados para avaliar a solução proposta, apresentando e discutindo os resultados obtidos, além de analisar a eficácia da arquitetura em diferentes cenários de uso.

Capítulo 6

Experimentos e Resultados

Este capítulo expõe os resultados computacionais obtidos para a avaliação da solução proposta, conforme descrito no Capítulo 5. Os experimentos conduzidos têm o objetivo de demonstrar a eficácia da solução proposta para tradução automática de Libras para o Português, além de validar as técnicas de pré-processamento e as arquiteturas empregadas. A análise dos resultados será realizada com base em métricas padronizadas, como WER e BLEU, além de outras métricas pertinentes que avaliam o desempenho e a precisão do sistema implementado.

6.1 Resultados

Os resultados são sumarizados na Tabela 6.1, nós podemos observar que a solução baseada em redes GRU obteve os melhores resultados nas métricas avaliadas, obtendo um WER mínimo de 21,62 e uma acurácia máxima de 92,68%, apesar da acurácia não ser muito usual nesse contexto, pois só captura as sentenças que são traduzidas perfeitamente, porém, ainda indica uma boa generalização do modelo. O experimento baseado em redes *transformers* foi inferior em todas as métricas avaliadas, isso pode ser explicado, em partes, pelo relativo baixo volume de dados, já que as redes *transformers* podem ter de centenas a bilhões de parâmetros treináveis, possuindo parâmetros suficientes para memorizar qualquer conjunto de dados pequeno, ou seja, sendo bastante suscetível ao problema de sobreajuste.

Tabela 6.1: Métricas dos modelos estudados.

Model	**Tipo**	***embedding***	**WER**	**BLEU**	**Acurácia**
Seq2seq	GRU	512	21,06	86,04%	87,21%
Seq2seq	GRU	768	21,62	95,84%	92,68%
Seq2seq	RNT	512	28,86	74,84%	70,09%

Nas Figuras 6.1 e 6.2 são exibidos os gráficos da evolução das métricas de avaliação WER e acurácia ao longo das épocas de treinamento, avaliadas sobre o conjunto de treinamento e de validação. Ambas as arquiteturas convergiram em um número pequeno de

épocas, frente ao limite de 50 épocas. A arquitetura Seq2Seq obteve melhores métricas e convergiu em apenas 5 épocas, após isso, a rede já entrou em sobreajuste, conforme indicada pela linha azul pontilhada. Já a arquitetura baseada em RNT convergiu mais lentamente, atingindo o seu melhor desempenho após 15 épocas.

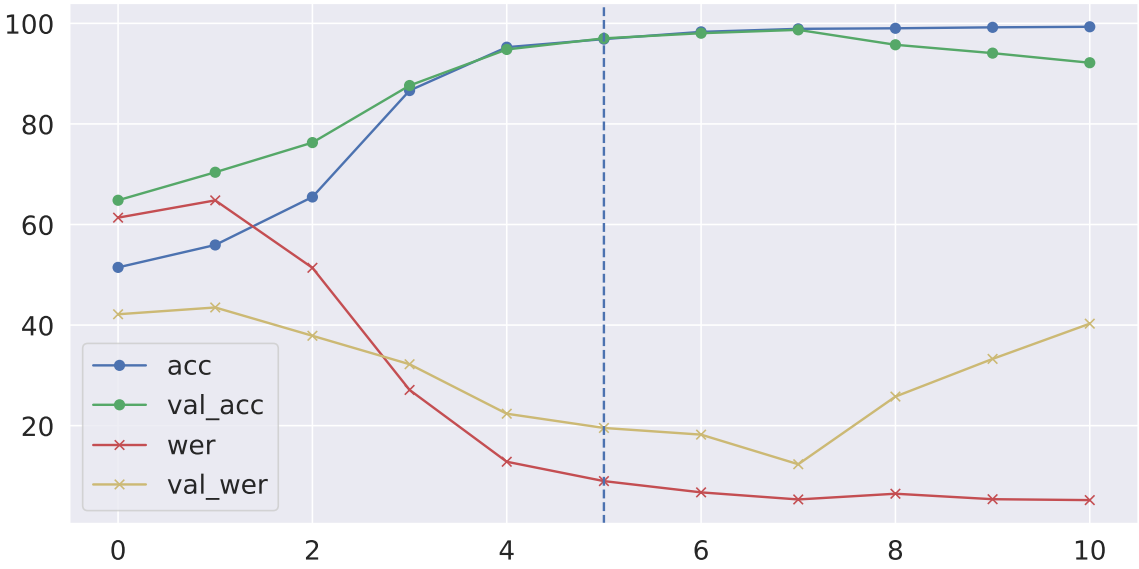

Figura 6.1: Métricas de treinamento e validação para arquitetura Seq2Seq. Fonte: Autor

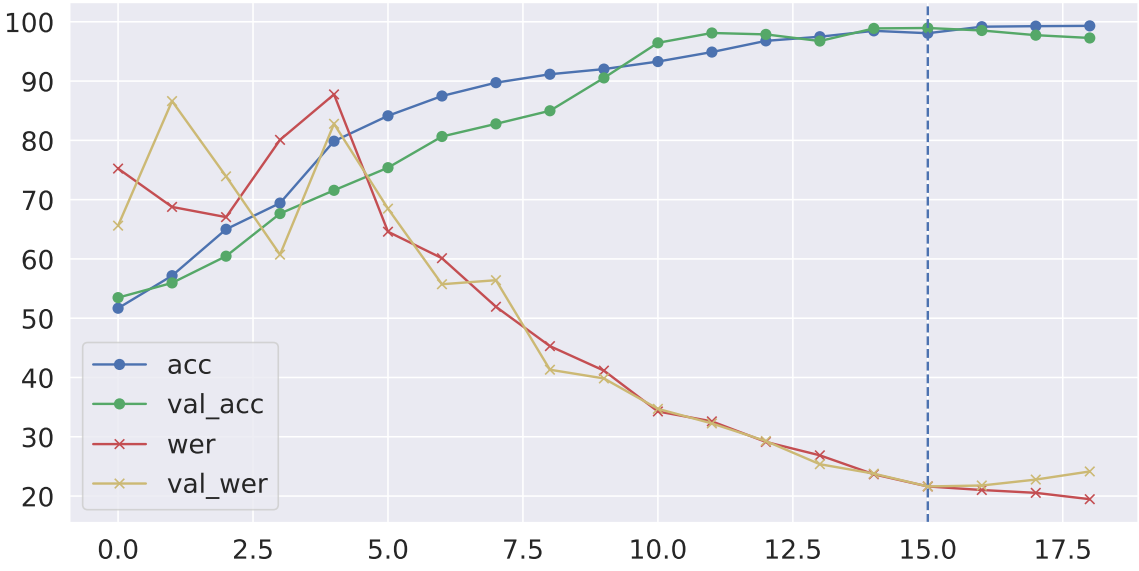

Figura 6.2: Métricas de treinamento e validação para arquitetura RNT. Fonte: Autor

Para verificar o impacto do uso da regularização nos resultados do modelo, também fizemos alguns testes usando algumas técnicas de regularização: L1 ou Lasso e L2 ou

Ridge, e *dropout*. Em relação ao *dropout*, foram aplicadas camadas individuais de *dropout* tanto no codificador quanto no decodificador. De acordo com a Tabela 6.2, o cenário que obteve o melhor resultado foi uma *dropout* de 0,5 para ambos os estágios. Sendo uma taxa de *dropout* relativamente alta, indicando que o modelo iria se beneficiar de um aumento do volume de dados.

Tabela 6.2: Acurácia para diferentes métodos de regularização.

Codificador	**Decodificador**	**WER**
0,2	0,2	27,78%
0,5	0,5	21,06%

Também foi constatado que o modelo apresenta melhores resultados usando dimensões de *embedding* maiores. Na maioria das vezes, esse parâmetro é escolhido empiricamente, por tentativa e erro; porém, o aumento da dimensão do *embedding* resulta em um maior custo computacional. As melhores métricas foram obtidas com uma dimensão de *embedding* de 768, valor usado por arquiteturas mais recentes como a BERT (Devlin et al. 2018).

A performance inferior do modelo baseado em RNT pode ser explicada, em grande parte, pela insuficiência do volume de dados disponível para o treinamento. Arquiteturas RNT são conhecidas por sua capacidade de capturar relações complexas em dados sequenciais, especialmente em tarefas de tradução e processamento de linguagem natural. No entanto, essa capacidade robusta vem acompanhada de um número extremamente alto de parâmetros a serem treinados, o que torna essas redes dependentes de grandes volumes de dados para atingir seu desempenho ideal.

Quando aplicamos RNT ao contexto de tradução automática de Libras para Português, a complexidade do problema se intensifica. Diferentemente de tarefas convencionais de tradução de texto, onde os *tokens* já estão presentes de forma linear e explícita, a tradução de Libras envolve a interpretação de dados visuais, que devem ser convertidos em uma representação vetorial antes de serem traduzidos para a língua-alvo. Isso significa que o modelo precisa capturar não apenas relações temporais entre os gestos, mas também variações espaciais, como a posição das mãos, expressões faciais e orientações corporais, o que exige uma grande quantidade de informações detalhadas.

Além disso, a natureza dos sinais em Libras, que variam em duração, movimento e significado contextual, requer uma modelagem temporal refinada, algo que os RNT podem realizar com eficiência, mas apenas quando o volume de dados é suficientemente grande para explorar essas dinâmicas. O treinamento de um modelo complexo, como o RNT, sem dados suficientes, resulta em uma rede incapaz de alinhar corretamente os gestos visuais com as correspondências textuais na língua-alvo, o que leva a uma performance inferior em métricas como WER e BLEU, bem como na acurácia geral da tradução.

Portanto, o desempenho limitado observado no modelo RNT não reflete a inadequação da arquitetura em si para a tarefa de tradução automática de Libras, mas sim a falta de dados adequados para explorar seu pleno potencial.

6.1.1 Impacto do Mecanismo de Atenção Visual

O mecanismo de atenção tem-se mostrado uma ferramenta fundamental em várias arquiteturas de redes neurais, especialmente em tarefas de tradução automática, onde a ordem e o contexto dos elementos de entrada são essenciais para gerar resultados precisos. No contexto deste trabalho, o uso de atenção visual no modelo de tradução automática de Libras para o Português proporcionou ganhos significativos em termos de qualidade da tradução, redução de erros e aumento da acurácia. A tabela a seguir resume os principais resultados com e sem o uso de atenção.

Tabela 6.3: Métricas dos modelos estudados.

Atenção	**WER**	**BLEU**	**Acurácia**
Sim	21,62	95,84%	92,68%
Não	40,77	75,10%	77,66%

A métrica WER é amplamente utilizada para avaliar sistemas de tradução automática e reconhecimento de fala, medindo a proporção de palavras incorretas, inseridas ou omitidas em relação à referência correta. Com o uso do mecanismo de atenção, o WER apresentou uma redução significativa, passando de 40,77% para 21,62%. Essa diminuição de quase 46,97% percentuais sugere que o mecanismo de atenção melhora a capacidade do modelo de focar nas partes mais relevantes da sequência de entrada.

A pontuação BLEU, outra métrica fundamental para avaliar a qualidade das traduções, também evidenciou ganhos notáveis. Sem atenção, o BLEU foi de 75,10%, indicando um desempenho mediano. Com o mecanismo de atenção, a pontuação aumentou para 95,84%, representando uma melhoria percentual de mais de 21,64%.

Sem a atenção, o modelo enfrenta dificuldades para capturar a estrutura temporal dos sinais em Libras, o que resulta em mais erros e menor qualidade de tradução. Ao permitir que o modelo direcione sua atenção para as partes cruciais do vídeo em momentos específicos, o mecanismo de atenção reduz essas falhas, capturando melhor os detalhes sutis dos gestos e suas relações contextuais.

Além disso, o uso de atenção não apenas melhora a precisão da tradução, mas também gera sentenças mais próximas da versão correta, tanto na estrutura gramatical quanto na escolha das palavras. Isso é especialmente relevante em tarefas de tradução de Libras, onde as diferenças na ordem das palavras e na gramática em relação ao Português são consideráveis. O mecanismo de atenção permite um alinhamento mais eficaz dos gestos visuais com o texto em Português, resultando em traduções mais fluentes e semanticamente precisas. Esse desempenho superior ocorre porque o mecanismo de atenção permite que o modelo foque nos “fonemas visuais” que compõem os sinais de Libras, como a posição das mãos, movimentos e expressões faciais que, juntos, formam cada gesto. Ao capturar e ponderar essas características visuais críticas, o modelo com atenção consegue identificar melhor as partes relevantes de cada sinal e relacioná-las com o contexto mais amplo da frase.

6.1.2 Refinamento com LLM

O *fine-tuning* do modelo LLaMA (Ver Figura 6.3) para tradução de glosa em Libras para o português resultou em uma pontuação BLEU de 64,36, um valor considerado alto, especialmente quando levado em conta o conjunto de avaliação utilizado, o corpus de validação do VLibras. Este corpus é conhecido por sua complexidade, contendo não apenas estruturas linguísticas comuns, mas também nuances específicas da Libras, como verbos direcionais, intensificadores, números e contextos que são característicos dessa língua visual. A presença desses elementos desafiadores torna a tarefa de tradução significativamente mais difícil, já que a gramática e as construções semânticas de Libras podem ser bastante diferentes do português.

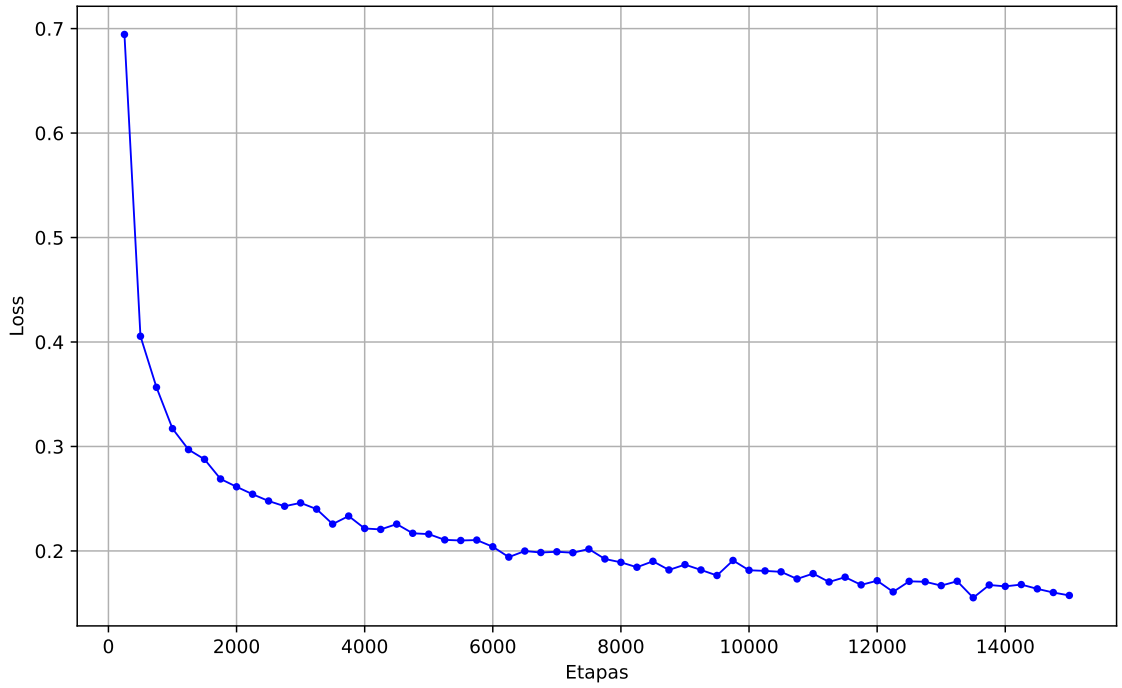

Figura 6.3: Evolução do *fine-tuning* do LLaMA-3B para tradução da glosa em português. Fonte: Autor

Nesse sentido, o BLEU de 64,36 sugere que o modelo foi capaz de lidar bem com essas nuances, capturando a estrutura, a semântica dos sinais e as diferenças contextuais de forma que a tradução para o português se aproxime muito das traduções de referência. A capacidade do LLaMA de preservar a essência semântica dos sinais, mesmo diante de desafios como verbos direcionais (que indicam o sujeito e o objeto através do movimento das mãos), intensificadores (que modificam a intensidade de uma ação ou estado), e outras construções específicas, demonstra sua eficácia em manter a integridade da comunicação entre as duas línguas.

Apesar do alto desempenho, o modelo LLaMA ainda apresenta limitações na tradução de glosas que se desviam significativamente da estrutura ou do padrão típico encontrado

Tabela 6.4: Exemplos de traduções de glosa para português usando o LLaMA adaptado.

Glosa	Português
JOGAR_FUTEBOL SENTIR DOENTE	Sinto-me doente depois de jogar futebol
MEU ALTURA_ESTATURA 1 VIRGULA 60 [PONTO]	Minha altura é 1,60.
EU ACIDENTE	Eu sofri um acidente
COMO 1S_AJUDAR_2S [INTERROGAÇÃO]	Como eu posso te ajudar?
NAO_TER ALERGIA	Não ter alergia
2S_AJUDAR_1S [INTERROGAÇÃO	Você me ajuda?

no corpus de treinamento, ou que foram reconhecidas incorretamente. Esse problema é agravado pela limitação do conjunto de dados, que é relativamente pequeno para capturar toda a diversidade linguística da Libras, especialmente considerando suas variações regionais e contextuais.

Tabela 6.5: Comparação da métrica BLEU entre o conjunto de validação generalista do VLibras e um conjunto de dados específico para o contexto da saúde.

Métrica	VLibras Validação	Contexto da Saúde
BLEU	64,36	53,58

Adicionalmente, o corpus de treinamento do VLibras, embora eficaz, é generalista e não foi projetado explicitamente para contextos específicos, como o da saúde, foco deste estudo. Essa ausência de especialização limita a eficácia do modelo em situações onde vocabulário e estruturas sintáticas específicas seriam essenciais para traduções precisas e contextualmente relevantes. Para avaliar o impacto dessa limitação, foi construído um conjunto de avaliação específico para o contexto da saúde. Os resultados dessa comparação são apresentados na Tabela 6.5, onde se observa uma redução de 16,75 pontos percentuais na métrica BLEU quando o modelo é avaliado com dados do contexto da saúde. Assim, um corpus especializado poderia potencialmente melhorar o desempenho do modelo, possibilitando uma tradução mais fiel às nuances e terminologias próprias de cada domínio.

Aplicando este modelo à saída do melhor modelo de tradução Libras-Português, obtemos uma pontuação BLEU de 92,77, representando uma variação mínima e um ganho prático de 0,09. Embora pequeno, esse ganho é esperado, pois a pontuação BLEU do primeiro estágio do tradutor já é bastante elevada, reduzindo o impacto visível de melhorias ou refinamentos adicionais. A tendência, no entanto, é que esse impacto aumente à medida que o vocabulário do corpus de treinamento se expande, tornando a tradução mais desafiadora e, assim, possibilitando que o uso do LLM gere um impacto positivo mais expressivo nas métricas.

6.2 Discussões Sobre o Experimento e Resultados

Neste trabalho, propomos uma metodologia de tradução contínua para Língua Brasileira de Sinais (Libras). Com uma metodologia inteiramente baseada em imagens ou sequências de imagens (vídeo), ou seja, não dependendo de hardwares de captura especiais tais como luvas ou sensores de profundidade, possuindo assim uma maior viabilidade para implantação em ambientes reais. Os resultados mostram um WER mínimo de 21,62% e um BLEU máximo de 92,68%, métricas calculadas sobre um conjunto de testes onde o intérprete não participou do processo de treinamento ou validação. Além disso, desenvolvemos um novo conjunto de dados em Libras com 10.500 vídeos específicos para o contexto de triagem clínica, visando apoiar o desenvolvimento de soluções que auxiliem na comunicação de pessoas surdas em ambientes de saúde e fomentar pesquisas adicionais nesse tema.

Os resultados computacionais indicaram a viabilidade da solução proposta; contudo, é importante destacar que a base de dados utilizada neste estudo é relativamente pequena para o escopo da tradução automática contínua em Libras. Com 10.500 vídeos em contexto clínico, a base proporciona uma fundação inicial e relevante para o desenvolvimento da metodologia proposta, especialmente por capturar o vocabulário e as expressões específicas desse contexto. A complexidade da Libras, que envolve estruturas gramaticais e contextos gestuais variados, sujeitos a diferenças regionais e de estilo dos sinalizadores, exige um volume de dados maior para capturar suas nuances linguísticas de forma abrangente. A limitação de dados restringe a capacidade do modelo de generalizar de forma robusta, o que pode impactar negativamente a performance quando aplicado em contextos diferentes dos presentes no conjunto de treinamento.

O tempo de inferência do modelo de tradução apresentou uma média de aproximadamente 3 segundos por tradução. Esse tempo inclui o processamento da entrada em Libras, a análise e interpretação do modelo e a geração da saída em português. Apesar de pequenas variações dependendo da complexidade do sinal e das condições de execução, a latência observada mantém-se dentro de um intervalo aceitável para aplicações em tempo real, garantindo uma experiência fluida e eficiente para os usuários.

Este trabalho teve seu escopo delimitado ao contexto da saúde, dado seu caráter essencial e visando tornar a investigação mais viável, uma vez que a criação de um conjunto de dados generalista demandaria um investimento considerável em tempo e recursos. No entanto, a metodologia e a solução propostas possuem adaptabilidade para outros domínios, como segurança, educação ou atendimento ao cliente, desde que se disponha de dados específicos e adequados para o treinamento. A flexibilidade da abordagem amplifica seu potencial de aplicação, permitindo facilitar a comunicação entre pessoas surdas e ouvintes em uma variedade de contextos, ampliando assim o impacto positivo da solução.

O capítulo a seguir apresenta as conclusões sobre o presente trabalho, como também as propostas de trabalhos futuros.

Capítulo 7

Conclusão

Neste trabalho, foi lapidada e validada a hipótese de que é possível realizar a tradução automática da Língua Brasileira de Sinais a partir de vídeos, ou de uma sequência de imagens digitais, para ser usada no contexto de aplicações sem um tradutor humano, usando como estudo de caso a situação de atendimento médico, na saúde. A solução proposta no trabalho tem importância no contexto social, principalmente no tocante às tecnologias assistivas e de inclusão, pois contribui para diminuir as barreiras de acessibilidade em consultas médicas enfrentadas pelos deficientes auditivos.

Como contribuição principal, foi desenvolvido um sistema completo, com diversos componentes tecnológicos e metodológicos que compõem a solução proposta, incluindo, além do sistema em si: a revisão das principais abordagens existentes para a tradução de línguas de sinais, o desenvolvimento de estratégias de tradução contínua, e a criação de bases de dados específicas para o domínio da saúde. Além disso, foi realizada uma avaliação rigorosa do sistema, em termos de acurácia e precisão na tradução dos sinais, utilizando métricas específicas para validar a capacidade da solução em oferecer um suporte eficiente e confiável em interações médicas.

Os resultados obtidos demonstram que, embora ainda existam desafios técnicos relacionados ao reconhecimento contínuo de sinais, é possível alcançar um nível de tradução adequado, pelo menos no contexto de saúde. A especialização do sistema nesse domínio contribuiu para a melhoria na precisão das traduções, facilitando a comunicação direta entre profissionais de saúde e pacientes surdos, promovendo um atendimento mais inclusivo e humanizado.

Conclui-se, portanto, que a tecnologia desenvolvida pode ser uma ferramenta valiosa no suporte à acessibilidade, com aplicação em serviços de saúde, reduzindo as barreiras de comunicação e possibilitando que pessoas com deficiência auditiva tenham um acesso mais equitativo a serviços essenciais. Apesar dos avanços, o trabalho reconhece que o sistema não substitui o papel fundamental dos intérpretes humanos, mas oferece uma alternativa importante em situações onde sua presença não é possível, garantindo a continuidade do atendimento e promovendo uma maior independência para a comunidade surda. Acreditamos que ainda precisa de mais estudos e desenvolvimentos para generalizar o sistema para situações diversas, do estudo de caso visto neste trabalho, o que será objeto de trabalho futuro. Ou seja, futuras pesquisas poderão expandir o escopo para outros contextos e aprimorar a capacidade de generalização do sistema, com vistas a ampliar ainda mais o impacto positivo da tecnologia no cotidiano das pessoas com

deficiência auditiva.

7.1 Propostas para Trabalhos Futuros

Como propostas para estudos futuros, são consideradas a simplificação dos modelos de tradução automática desenvolvidos é essencial para torná-los aplicáveis em dispositivos móveis, garantindo acessibilidade e mobilidade aos usuários. A complexidade dos modelos utilizados em servidores de alto desempenho não é adequada para dispositivos com recursos limitados, como smartphones ou tablets. Portanto, uma linha futura de pesquisa importante envolve a redução da complexidade computacional dos algoritmos, seja por meio da otimização dos modelos, poda de redes neurais, quantização dos pesos, ou uso de técnicas de aprendizagem leve. A portabilidade e a eficiência energética também são aspectos críticos a serem considerados para permitir que a solução seja utilizada amplamente, promovendo acessibilidade a um público diversificado.

Além disso, a prototipação de uma ferramenta funcional que incorpore a metodologia de tradução automática proposta. Esse protótipo deve ser desenvolvido com uma interface de usuário acessível, que facilite a interação tanto para pessoas surdas quanto para ouvintes. Tal ferramenta poderia ser implementada como um aplicativo ou plataforma web, que faça uso das estratégias de reconhecimento contínuo de sinais em tempo real. A construção desse protótipo permitirá não apenas a avaliação técnica da metodologia, mas também testes preliminares que podem guiar melhorias na interface e na experiência do usuário.

Também é considerada um eventual integração da solução proposta com o sistema "Meu SUS Digital"[1], com o objetivo de expandir a acessibilidade e garantir que pacientes surdos possam utilizar a tradução automática diretamente no ambiente de serviços públicos de saúde. A integração com a plataforma permitiria que o sistema de tradução automática fosse acessado de maneira conveniente por usuários durante suas interações com o sistema de saúde, viabilizando consultas remotas e presenciais com suporte à comunicação por Libras, além de melhorar o acesso a informações de saúde personalizadas.

Finalmente, a validação da metodologia desenvolvida por meio de testes com usuários em um ambiente real representa um estágio essencial para avaliar a eficácia prática do sistema proposto. A coleta de dados a partir de testes com usuários reais permitirá medir a acurácia, a usabilidade, e a aceitação social da ferramenta em contextos cotidianos. Esses testes também poderão identificar desafios e limitações que não são detectáveis em um ambiente controlado, como variações naturais na expressão da língua de sinais, diferentes condições de iluminação, e outros fatores que afetam a precisão do reconhecimento dos sinais. Além disso, o *feedback* dos usuários fornecerá informações valiosas para adequar a solução às necessidades reais da comunidade surda, assegurando que o desenvolvimento atenda plenamente às expectativas dos usuários finais.

[1] O Meu SUS Digital, antigo Conecte SUS, é um aplicativo e uma plataforma web que permite o acesso a diversos serviços de saúde do Sistema Único de Saúde (SUS), incluindo a telessaúde

Referências Bibliográficas

Abreu, João Gabriel, João Marcelo Teixeira, Lucas Silva Figueiredo & Veronica Teichrieb (2016), Evaluating sign language recognition using the myo armband, *em* '2016 XVIII Symposium on Virtual and Augmented Reality (SVR)', pp. 64–70.

Adaloglou, Nikolas, Theocharis Chatzis, Ilias Papastratis, Andreas Stergioulas, Georgios Th. Papadopoulos, Vassia Zacharopoulou, George J. Xydopoulos, Klimnis Atzakas, Dimitris Papazachariou & Petros Daras (2020), 'A comprehensive study on sign language recognition methods', *CoRR* **abs/2007.12530**.
URL: *https://arxiv.org/abs/2007.12530*

Akmeliawati, Rini, Melanie Po-Leen Ooi & Ye Chow Kuang (2007), Real-time malaysian sign language translation using colour segmentation and neural network, *em* '2007 IEEE Instrumentation & Measurement Technology Conference IMTC 2007', IEEE.
URL: *https://doi.org/10.1109/imtc.2007.379311*

Araujo, Tiago, Felipe Ferreira, Danilo Silva, Leonardo Oliveira, Eduardo Falcão, Vandhuy Martins, Igor Portela, Yurika Nóbrega, Hozana Lima, Guido Souza Filho, Tatiana Tavares & Alexandre Duarte (2014), 'An approach to generate and embed sign language video tracks into multimedia contents', *Information Sciences* **281**, 762–.

Arnab, Anurag, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lucic & Cordelia Schmid (2021), 'Vivit: A video vision transformer', *CoRR* **abs/2103.15691**.
URL: *https://arxiv.org/abs/2103.15691*

Bahdanau, Dzmitry, Kyunghyun Cho & Yoshua Bengio (2015), Neural machine translation by jointly learning to align and translate, *em* Y.Bengio & Y.LeCun, eds., '3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings'.
URL: *http://arxiv.org/abs/1409.0473*

Bessa Carneiro, S., E. D. F. De M. Santos, T. M. De A. Barbosa, J. O. Ferreira, S. G. Soares Alcalá & A. F. Da Rocha (2016), Static gestures recognition for brazilian sign language with kinect sensor, *em* '2016 IEEE SENSORS', pp. 1–3.

Bianco, Pedro Dal, Gastón Ríos, Franco Ronchetti, Facundo Quiroga, Oscar Stanchi, Waldo Hasperué & Alejandro Rosete (2022), 'Lsa-t: The first continuous argentinian sign language dataset for sign language translation'.
URL: *https://arxiv.org/abs/2211.15481*

Binh, Nguyen Dang & Toshiaki Ejima (2005), Real-time malaysian sign language translation using colour segmentation and neural network, *em* 'Proc. ICGST Int. Conf. Graph. Vision Image Process', pp. 1–6.
URL: *http://www.math.hcmuns.edu.vn/p̃tbao/LVTN/2003/hand_Gesture P1150535210.pdf*

Brown, Tom B., Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever & Dario Amodei (2020), 'Language models are few-shot learners', *CoRR* **abs/2005.14165**.
URL: *https://arxiv.org/abs/2005.14165*

Buttussi, Fabio, Luca Chittaro & Marco Coppo (2007), Using web3d technologies for visualization and search of signs in an international sign language dictionary, Vol. 2007, pp. 61–70.

Camgoz, Necati Cihan, Oscar Koller, Simon Hadfield & Richard Bowden (2020*a*), Multi-channel transformers for multi-articulatory sign language translation, *em* A.Bartoli & A.Fusiello, eds., 'Computer Vision – ECCV 2020 Workshops', Springer International Publishing, Cham, pp. 301–319.

Camgöz, Necati Cihan, Oscar Koller, Simon Hadfield & Richard Bowden (2020*b*), 'Sign language transformers: Joint end-to-end sign language recognition and translation', *CoRR* **abs/2003.13830**.
URL: *https://arxiv.org/abs/2003.13830*

Camgoz, Necati Cihan, Simon Hadfield, Oscar Koller, Hermann Ney & Richard Bowden (2018*a*), Neural sign language translation, *em* '2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition', pp. 7784–7793.

Camgoz, Necati, Simon Hadfield, Oscar Koller, Hermann Ney & Richard Bowden (2018*b*), Neural sign language translation.

Cao, Zhe, Gines Hidalgo, Tomas Simon, Shih-En Wei & Yaser Sheikh (2018), OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields, *em* 'arXiv preprint arXiv:1812.08008'.

Carreira, João & Andrew Zisserman (2017), 'Quo vadis, action recognition? A new model and the kinetics dataset', *CoRR* **abs/1705.07750**.
URL: *http://arxiv.org/abs/1705.07750*

Cerna, Lourdes Ramirez, Edwin Escobedo Cardenas, Dayse Garcia Miranda, David Menotti & Guillermo Camara-Chavez (2021), 'A multimodal libras-ufop brazilian sign language dataset of minimal pairs using a microsoft kinect sensor', *Expert Systems*

with Applications **167**, 114179.
URL: *https://www.sciencedirect.com/science/article/pii/S0957417420309143*

Chaudhari, Sneha, Varun Mithal, Gungor Polatkan & Rohan Ramanath (2021), 'An attentive survey of attention models', *ACM Trans. Intell. Syst. Technol.* **12**(5).
URL: *https://doi.org/10.1145/3465055*

Cheng, De, Yihong Gong, Sanping Zhou, Jinjun Wang & Nanning Zheng (2016), Person re-identification by multi-channel parts-based cnn with improved triplet loss function, *em* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)'.

Cheok, Ming Jin, Zaid Omar & Mohamed Hisham Jaward (2017), 'A review of hand gesture and sign language recognition techniques', *International Journal of Machine Learning and Cybernetics* .
URL: *https://doi.org/10.1007/s13042-017-0705-5*

Chuan, C., E. Regina & C. Guardino (2014), American sign language recognition using leap motion sensor, *em* '2014 13th International Conference on Machine Learning and Applications', pp. 541–544.

Chung, Junyoung, Çaglar Gülçehre, KyungHyun Cho & Yoshua Bengio (2014), 'Empirical evaluation of gated recurrent neural networks on sequence modeling', *CoRR* **abs/1412.3555**.
URL: *http://arxiv.org/abs/1412.3555*

Cooper, Helen, Nicolas Pugeault & Richard Bowden (2011), Reading the signs: A video based sign dictionary, *em* '2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)', IEEE.
URL: *https://doi.org/10.1109/iccvw.2011.6130349*

de Albuquerque, Dilainne Daniel (2021), Modelagem e desenvolvimento de uma base de dados para tradução automática de conteúdos digitais para língua brasileira de sinais, Dissertação de mestrado, Universidade Federal da Paraíba, Brasil.
URL: *https://repositorio.ufpb.br/jspui/handle/123456789/31303*

de Araújo, Tiago Maritan Ugulino, Felipe Lacet S. Ferreira, Danilo Assis Nobre dos S. Silva, Felipe Hermínio Lemos, Gutenberg Pessoa Botelho Neto, Derzu Omaia, Guido Lemos de Souza Filho & Tatiana A. Tavares (2012), 'Automatic generation of brazilian sign language windows for digital tv systems', *Journal of the Brazilian Computer Society* **19**, 107–125.

De Coster, Mathieu, Mieke Van Herreweghe & Joni Dambre (2020), Sign language recognition with transformer networks, *em* 'PROCEEDINGS OF THE 12TH INTERNATIONAL CONFERENCE ON LANGUAGE RESOURCES AND EVALUATION (LREC 2020)', European Language Resources Association (ELRA), pp. 6018–6024.
URL: *http://www.lrec-conf.org/proceedings/lrec2020/LREC-2020.pdf*

de Souza, Maria Fernanda Neves Silveira, Amanda Miranda Brito Araújo, Luiza Fernandes Fonseca Sandes, Daniel Antunes Freitas, Wellington Danilo Soares, Raquel Schwenck de Mello Vianna & Árlen Almeida Duarte de Sousa (2017), 'Principais dificuldades e obstáculos enfrentados pela comunidade surda no acesso à saúde: uma revisão integrativa de literatura', *Revista CEFAC* **19**(3), 395–405.
URL: *https://doi.org/10.1590/1982-0216201719317116*

Devlin, Jacob, Ming-Wei Chang, Kenton Lee & Kristina Toutanova (2018), 'BERT: pre-training of deep bidirectional transformers for language understanding', *CoRR* **abs/1810.04805**.
URL: *http://arxiv.org/abs/1810.04805*

Dias, Andrezza, Cinthya Coutinho, Deborah Gaspar, Leticia Moeller & Marcelo Mamede (2017), 'Libras na formação médica: possibilidade de quebra da barreira comunicativa e melhora na relação médico-paciente surdo', *Revista de Medicina* **96**, 209.

Dizeu, Liliane Correia Toscano de Brito & Sueli Aparecida Caporali (2005), 'A língua de sinais constituindo o surdo como sujeito', *Educação & Sociedade* **26**, 583 – 597.
URL: *http://www.scielo.br/scielo.php?script=sci*$_a$*rttextpid* $=$ *S*0101 $-$ 73302005000200014*nrm* $=$ *iso*

Dosovitskiy, Alexey, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit & Neil Houlsby (2020), 'An image is worth 16x16 words: Transformers for image recognition at scale', *CoRR* **abs/2010.11929**.
URL: *https://arxiv.org/abs/2010.11929*

Felipe, T. A. (2006), *Os processos de formação de palavras na Libras*, Vol. 7, TD-Educação Temática Digital, Campinas, SP.

Felipe, Tanya Amara (2008), 'Os processos de formação de palavra na libras', *ETD - Educação Temática Digital* **7**(2), 200.
URL: *https://doi.org/10.20396/etd.v7i2.803*

Galicia, R., O. Carranza, E. D. Jiménez & G. E. Rivera (2015), Mexican sign language recognition using movement sensor, *em* '2015 IEEE 24th International Symposium on Industrial Electronics (ISIE)', pp. 573–578.

Gardner, M.W & S.R Dorling (1998), 'Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences', *Atmospheric Environment* **32**(14), 2627–2636.
URL: *https://www.sciencedirect.com/science/article/pii/S1352231097004470*

Goodfellow, Ian J., Yoshua Bengio & Aaron Courville (2016), *Deep Learning*, MIT Press, Cambridge, MA, USA. http://www.deeplearningbook.org.

Goyal, Sakshi, Ishita Sharma & Shanu Sharma (2013), 'Sign language recognition system for deaf and dumb people', *International Journal of Engineering Research Technology* **2**(4).

Graves, Alex, Santiago Fernández, Faustino Gomez & Jürgen Schmidhuber (2006), Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks, *em* 'Proceedings of the 23rd International Conference on Machine Learning', ICML '06, Association for Computing Machinery, New York, NY, USA, p. 369–376.
URL: *https://doi.org/10.1145/1143844.1143891*

Guo, Dan, Wengang Zhou, Houqiang Li & Meng Wang (2018), 'Hierarchical lstm for sign language translation', *Proceedings of the AAAI Conference on Artificial Intelligence* **32**(1).
URL: *https://ojs.aaai.org/index.php/AAAI/article/view/12235*

Haykin, Simon S. (2009), *Neural networks and learning machines*, third[a] edição, Pearson Education, Upper Saddle River, NJ.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren & Jian Sun (2015), 'Deep residual learning for image recognition', *CoRR* **abs/1512.03385**.
URL: *http://arxiv.org/abs/1512.03385*

Herreweghe, Mieke Van, Myriam Vermeerbergen, Eline Demey, Hannes De Durpel, Hilde Nyffels & Sam Verstraete (2015), Het corpus vgt. een digitaal open access corpus van videos and annotaties van vlaamse gebarentaal, ontwikkeld aan de universiteit gent ism ku leuven.

Hochreiter, Sepp & Jürgen Schmidhuber (1997), 'Long short-term memory', *Neural Comput.* **9**(8), 1735–1780.
URL: *http://dx.doi.org/10.1162/neco.1997.9.8.1735*

Hu, Jie, Li Shen & Gang Sun (2017), 'Squeeze-and-excitation networks', *CoRR* **abs/1709.01507**.
URL: *http://arxiv.org/abs/1709.01507*

Huang, J., W. Zhou, H. Li & W. Li (2019), 'Attention-based 3d-cnns for large-vocabulary sign language recognition', *IEEE Transactions on Circuits and Systems for Video Technology* **29**(9), 2822–2832.

Huang, Jie, Wengang Zhou, Qilin Zhang, Houqiang Li & Weiping Li (2018), 'Video-based sign language recognition without temporal segmentation', *CoRR* **abs/1801.10111**.
URL: *http://arxiv.org/abs/1801.10111*

Huang, Zhiheng, Wei Xu & Kai Yu (2015), 'Bidirectional LSTM-CRF models for sequence tagging', *CoRR* **abs/1508.01991**.
URL: *http://arxiv.org/abs/1508.01991*

Huenerfauth, Matt (2004), 'A multi-path architecture for machine translation of english text into american sign language animation'.

Huenerfauth, Matt (2008), 'Generating american sign language animation: overcoming misconceptions and technical challenges', *Universal Access in the Information Society* **6**, 419–434.

Jani, A. B., N. A. Kotak & A. K. Roy (2018), Sensor based hand gesture recognition system for english alphabets used in sign language of deaf-mute people, *em* '2018 IEEE SENSORS', pp. 1–4.

Jiang, Songyao, Bin Sun, Lichen Wang, Yue Bai, Kunpeng Li & Yun Fu (2021), 'Skeleton aware multi-modal sign language recognition', *CoRR* **abs/2103.08833**.
URL: *https://arxiv.org/abs/2103.08833*

Karnopp, Lodenir Becker (1999), 'Aquisição fonológica na língua brasileira de sinais: estudo longitudinal de uma criança surda'.

Kau, L., W. Su, P. Yu & S. Wei (2015), A real-time portable sign language translation system, *em* '2015 IEEE 58th International Midwest Symposium on Circuits and Systems (MWSCAS)', pp. 1–4.

Kaya, F., A. F. Tuncer & Ş. K. Yildiz (2018), Detection of the turkish sign language alphabet with strain sensor based data glove, *em* '2018 26th Signal Processing and Communications Applications Conference (SIU)', pp. 1–4.

Kelly, Daniel, Jane Reilly Delannoy, John Mc Donald & Charles Markham (2009), A framework for continuous multimodal sign language recognition, *em* 'Proceedings of the 2009 International Conference on Multimodal Interfaces', ICMI-MLMI '09, Association for Computing Machinery, New York, NY, USA, p. 351–358.
URL: *https://doi.org/10.1145/1647314.1647387*

Kingma, Diederik P. & Jimmy Ba (2014), 'Adam: A method for stochastic optimization', *CoRR* **abs/1412.6980**.
URL: *http://arxiv.org/abs/1412.6980*

Kishore, PVV, MVD Prasad, D Anil Kumar & ASCS Sastry (2016), Optical flow hand tracking and active contour hand shape features for continuous sign language recognition with artificial neural networks, *em* '2016 IEEE 6th international conference on advanced computing (IACC)', IEEE, pp. 346–351.

Ko, Sang-Ki, Chang Jo Kim, Hyedong Jung & Choongsang Cho (2019), 'Neural sign language translation based on human keypoint estimation', *Applied Sciences* **9**(13).
URL: *https://www.mdpi.com/2076-3417/9/13/2683*

Koller, Oscar, Jens Forster & Hermann Ney (2015), 'Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers', *Computer Vision and Image Understanding* **141**, 108–125.

Koller, Oscar, Sepehr Zargaran, Hermann Ney & Richard Bowden (2016), Deep sign: Hybrid cnn-hmm for continuous sign language recognition.

Kumar, Pradeep, Himaanshu Gauba, Partha Pratim Roy & Debi Prosad Dogra (2017), 'Coupled hmm-based multi-sensor data fusion for sign language recognition', *Pattern Recognition Letters* **86**, 1–8.
URL: *https://www.sciencedirect.com/science/article/pii/S0167865516303518*

Kumar, Pradeep, Himaanshu Gauba, Partha Pratim Roy & Debi Prosad Dogra (2017), 'A multimodal framework for sensor based sign language recognition', *Neurocomputing* **259**, 21–38. Multimodal Media Data Understanding and Analytics.
URL: *https://www.sciencedirect.com/science/article/pii/S092523121730262X*

LeCun, Yann, Y. Bengio & Geoffrey Hinton (2015), 'Deep learning', *Nature* **521**, 436–44.
URL: *http://dx.doi.org/doi:10.1038/nature14539*

Lim, Kian Ming, Alan WC Tan & Shing Chiang Tan (2016), 'Block-based histogram of optical flow for isolated sign language recognition', *Journal of Visual Communication and Image Representation* **40**, 538–545.

Lugaresi, Camillo, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, Wan-Teh Chang, Wei Hua, Manfred Georg & Matthias Grundmann (2019), 'Mediapipe: A framework for building perception pipelines', *CoRR* **abs/1906.08172**.
URL: *http://arxiv.org/abs/1906.08172*

Luong, Minh-Thang, Hieu Pham & Christopher D. Manning (2015), 'Effective approaches to attention-based neural machine translation', *CoRR* **abs/1508.04025**.
URL: *http://arxiv.org/abs/1508.04025*

López-Ludeña, Verónica, Carlos Morcillo, Juan Carlos López, E. Ferreiro, Javier Ferreiros & Ruben Hernandez (2014), 'Methodology for developing an advanced communications system for the deaf in a new domain', *Knowledge-Based Systems* **56**, 240–252.

López-Ludeña, Verónica, Carlos Morcillo, Juan Carlos López, Roberto Barra-Chicote, Ricardo Cordoba & Ruben Hernandez (2014), 'Translating bus information into sign language for deaf people', *Engineering Applications of Artificial Intelligence* **32**.

Malaia, Evie A., Joshua D. Borneman & Ronnie B. Wilbur (2018), 'Information transfer capacity of articulators in american sign language', *Language and Speech* **61**, 112 – 97.

Marr, David (1982), *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*, Henry Holt and Co., Inc., New York, NY, USA.

Masood, Sarfaraz, Adhyan Srivastava, Harish Thuwal & Musheer Ahmad (2018), *Real-Time Sign Language Gesture (Word) Recognition from Video Sequences Using CNN and RNN*, pp. 623–632.

Morrissey, Sara & Andy Way (2013), 'Manual labour: Tackling machine translation for sign languages', *Machine Translation* **27**.

Moryossef, Amit, Ioannis Tsochantaridis, Roee Aharoni, Sarah Ebling & Srini Narayanan (2020), 'Real-time sign language detection using human pose estimation', *CoRR* **abs/2008.04637**.
URL: *https://arxiv.org/abs/2008.04637*

Papineni, Kishore, Salim Roukos, Todd Ward & Wei-Jing Zhu (2002), Bleu: a method for automatic evaluation of machine translation, *em* 'Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics', Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, pp. 311–318.
URL: *https://aclanthology.org/P02-1040*

Paszke, Adam, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai & Soumith Chintala (2019), 'Pytorch: An imperative style, high-performance deep learning library', *CoRR* **abs/1912.01703**.
URL: *http://arxiv.org/abs/1912.01703*

Pigou, Lionel, Sander Dieleman, Pieter-Jan Kindermans & Benjamin Schrauwen (2015), Sign language recognition using convolutional neural networks, *em* 'Computer Vision - ECCV 2014 Workshops', Springer International Publishing, pp. 572–578.
URL: *https://doi.org/10.1007/978-3-319-16178-5_40*

Radford, Alec, Jeff Wu, Rewon Child, David Luan, Dario Amodei & Ilya Sutskever (2019*a*), Language models are unsupervised multitask learners.

Radford, Alec, Jeff Wu, Rewon Child, David Luan, Dario Amodei & Ilya Sutskever (2019*b*), 'Language models are unsupervised multitask learners'.

Ramos, Clélia Regina (2004), 'Libras: A língua de sinais dos surdos brasileiros', *Disponível para download na página da Ediotra Arara Azul: http://www. editora-arara-azul. com. br/pdf/artigo2. pdf* .

Ranftl, René, Alexey Bochkovskiy & Vladlen Koltun (2021), 'Vision transformers for dense prediction', *CoRR* **abs/2103.13413**.
URL: *https://arxiv.org/abs/2103.13413*

Redmon, Joseph, Santosh Kumar Divvala, Ross B. Girshick & Ali Farhadi (2015), 'You only look once: Unified, real-time object detection', *CoRR* **abs/1506.02640**.
URL: *http://arxiv.org/abs/1506.02640*

Rodriguez, Jefferson, Juan Chacon, Edgar Rangel, Luis Guayacan, Claudia Hernandez, Luisa Hernandez & Fabio Martinez (2020), Understanding motion in sign language: A new structured translation dataset, *em* 'Proceedings of the Asian Conference on Computer Vision (ACCV)'.

Secco, Rosemeire & Maicon Herverton Lino Ferreira da Silva Barros (2009), Ambiente interativo para aprendizagem em libras gestual e escrita.

Sherstinsky, Alex (2018), 'Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network', *CoRR* **abs/1808.03314**.
URL: *http://arxiv.org/abs/1808.03314*

Shoaib, Umar, Nadeem Ahmad, Paolo Prinetto & G. Tiotto (2013), 'Integrating multiwordnet with italian sign language lexical resources', *Expert Systems with Applications* .

Simonyan, Karen & Andrew Zisserman (2014), 'Two-Stream Convolutional Networks for Action Recognition in Videos', *arXiv e-prints* p. arXiv:1406.2199.

Souza, Robson, José De Martino, Janice Temoteo & Ivani Rodrigues (2021), 'Automatic recognition of continuous signing of brazilian sign language for medical interview', pp. 41–46.

Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever & Ruslan Salakhutdinov (2014), 'Dropout: A simple way to prevent neural networks from overfitting', *Journal of Machine Learning Research* **15**(56), 1929–1958.
URL: *http://jmlr.org/papers/v15/srivastava14a.html*

Stokoe, William (2005), 'Sign language structure: An outline of the visual communication systems of the american deaf', *Journal of deaf studies and deaf education* **10**, 3–37.

Stokoe, William C., Jr. (2005), 'Sign Language Structure: An Outline of the Visual Communication Systems of the American Deaf', *The Journal of Deaf Studies and Deaf Education* **10**(1), 3–37.
URL: *https://doi.org/10.1093/deafed/eni001*

Stoll, Stephanie, Necati Cihan Camgoz, Simon Hadfield & Richard Bowden (2018), Sign language production using neural machine translation and generative adversarial networks, *em* 'Proceedings of the British Conference on Machine Vision (BMVC)', BMVA Press, Newcastle, UK.
URL: *http://personal.ee.surrey.ac.uk/Personal/S.Hadfield/papers/Stoll18.pdf*

Sutskever, Ilya, Oriol Vinyals & Quoc V. Le (2014), 'Sequence to sequence learning with neural networks', *CoRR* **abs/1409.3215**.
URL: *http://arxiv.org/abs/1409.3215*

Trockman, Asher & J. Zico Kolter (2022), 'Patches are all you need?', *CoRR* **abs/2201.09792**.
URL: *https://arxiv.org/abs/2201.09792*

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser & Illia Polosukhin (2017), 'Attention is all you need', *CoRR* **abs/1706.03762**.
URL: *http://arxiv.org/abs/1706.03762*

Webster, Jonathan & Chunyu Kit (1992), Tokenization as the initial phase in nlp, pp. 1106–1110.

Wei, C., W. Zhou, J. Pu & H. Li (2019), Deep grammatical multi-classifier for continuous sign language recognition, *em* '2019 IEEE Fifth International Conference on Multimedia Big Data (BigMM)', pp. 435–442.

WHO, World Health Organization (2021), *World report on hearing*, World Health Organization.

Wu, J., L. Sun & R. Jafari (2016), 'A wearable system for recognizing american sign language in real-time using imu and surface emg sensors', *IEEE Journal of Biomedical and Health Informatics* **20**(5), 1281–1290.

Xu, Kelvin, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel & Yoshua Bengio (2015), Show, attend and tell: Neural image caption generation with visual attention, *em* F.Bach & D.Blei, eds., 'Proceedings of the 32nd International Conference on Machine Learning', Vol. 37 de *Proceedings of Machine Learning Research*, PMLR, Lille, France, pp. 2048–2057.
URL: *https://proceedings.mlr.press/v37/xuc15.html*

Yang, Zichao, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola & Eduard Hovy (2016), Hierarchical attention networks for document classification, pp. 1480–1489.

Yauri Vidalón, José Elías & José Mario De Martino (2016), Brazilian sign language recognition using kinect, *em* G.Hua & H.Jégou, eds., 'Computer Vision – ECCV 2016 Workshops', Springer International Publishing, Cham, pp. 391–402.

Yauri Vidalón, José Elías & José Mario De Martino (2015), 'Continuous sign recognition of brazilian sign language in a healthcare setting', *Journal of Communication and Information Systems* **30**(1).
URL: *https://jcis.sbrt.org.br/jcis/article/view/99*

Yin, Fang, Xiujuan Chai & Xilin Chen (2016), Iterative reference driven metric learning for signer independent isolated sign language recognition, Vol. 9911, pp. 434–450.

Yin, Kayo (2020), 'Sign language translation with transformers', *CoRR* **abs/2004.00588**.
URL: *https://arxiv.org/abs/2004.00588*

Yuan, Li, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Francis E. H. Tay, Jiashi Feng & Shuicheng Yan (2021), 'Tokens-to-token vit: Training vision transformers from scratch on imagenet', *CoRR* **abs/2101.11986**.
URL: *https://arxiv.org/abs/2101.11986*

Zhang, Han, Ian Goodfellow, Dimitris Metaxas & Augustus Odena (2018), 'Self-attention generative adversarial networks'.
URL: *https://arxiv.org/abs/1805.08318*

Zhang, J., W. Zhou, C. Xie, J. Pu & H. Li (2016), Chinese sign language recognition with adaptive hmm, *em* '2016 IEEE International Conference on Multimedia and Expo (ICME)', pp. 1–6.