



Universidade Federal da Paraíba
Programa de Pós-Graduação em Informática

Sincronismo em língua de sinais: Uma
investigação de técnicas para redução da
dissincronia entre o Português e a
tradução automática para Língua
Brasileira de Sinais.

Richelieu Ramos de Andrade Costa

João Pessoa - PB

2024



Universidade Federal da Paraíba
Programa de Pós-Graduação em Informática

Richelieu Ramos de Andrade Costa

Sincronismo em língua de sinais: Uma investigação de técnicas para redução da dissincronia entre o Português e a tradução automática para Língua Brasileira de Sinais.

Dissertação apresentada à Universidade Federal da Paraíba como parte dos requisitos para a obtenção do título de Mestre em Informática.

Orientador: Guido Lemos de Souza Filho

João Pessoa - PB

2024

Catálogo na publicação
Seção de Catalogação e Classificação

C838s Costa, Richelieu Ramos de Andrade.
Sincronismo em língua de sinais : uma investigação de técnicas para redução da dissincronia entre o Português e a tradução automática para Língua Brasileira de Sinais. / Richelieu Ramos de Andrade Costa. - João Pessoa, 2024.
62 f.

Orientação: Guido Lemos de Souza Filho.
Dissertação (Mestrado) - UFPB/CI.

1. Tradução automática - Língua de Sinais. 2. Fluxo de vídeo. 3. Avatar 3D. 4. Acessibilidade. I. Souza Filho, Guido Lemos de. II. Título.

UFPB/BC

CDU 81'322.4(043)



UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA



Ata da Sessão Pública de Defesa de Dissertação de Mestrado de Richelieu Ramos de Andrade Costa, candidato ao título de Mestre em Informática na área de Sistemas de Computação, realizada em 30 de agosto de 2024.

Aos trinta dias do mês de agosto do ano de dois mil e vinte e quatro, às dezesseis horas, no Centro de Informática da Universidade Federal da Paraíba, reuniram-se os membros da Banca Examinadora constituída para julgar o Trabalho Final do discente Richelieu Ramos de Andrade Costa, vinculado a esta Universidade sob a matrícula nº 20221004760, candidato ao grau de Mestre em Informática, na área de “*Sistemas de Computação*”, na linha de pesquisa “*Computação Distribuída*”, do Programa de Pós-Graduação em Informática. A comissão examinadora foi composta pelos professores: Guido Lemos de Souza Filho, Orientador e Presidente da banca; Tiago Maritan Ugulino de Araujo, Examinador Interno; Cícero Inácio da Silva, Examinador Externo à Instituição. Dando início aos trabalhos, o Presidente da Banca cumprimentou os presentes, comunicou a finalidade da reunião e passou a palavra ao candidato para que ele fizesse a exposição oral do trabalho de dissertação intitulado “**Sincronismo em Língua de Sinais: Uma investigação de técnicas para redução da dissincronia entre o Português e a tradução automática para Língua Brasileira de Sinais**”. Concluída a exposição, o candidato foi arguido pela Banca Examinadora que emitiu o seguinte parecer: “**aprovado**”. Do ocorrido, eu, Gilberto Farias de Sousa Filho, Coordenador do Programa de Pós-Graduação em Informática, lavrei a presente ata que vai assinada por mim e pelos membros da Banca Examinadora. João Pessoa, 30 de agosto de 2024.

Gilberto Farias de Sousa Filho
Coordenador do Programa de Pós-Graduação em Informática

Prof. Dr. Guido Lemos de Souza Filho
Orientador (PPGI-UFPB)

Documento assinado digitalmente
gov.br GUIDO LEMOS DE SOUZA FILHO
Data: 28/10/2024 10:54:38-0300
Verifique em <https://validar.iti.gov.br>

Prof. Dr. Tiago Maritan Ugulino de Araujo
Examinador Interno (PPGI-UFPB)

Documento assinado digitalmente
gov.br TIAGO MARITAN UGULINO DE ARAUJO
Data: 25/10/2024 09:29:40-0300
Verifique em <https://validar.iti.gov.br>

Prof. Dr. Cícero Inácio da Silva
Examinador Externo à Instituição (UNIFESP)

Documento assinado digitalmente
gov.br CICERO INACIO DA SILVA
Data: 25/10/2024 20:35:32-0300
Verifique em <https://validar.iti.gov.br>

Abstract

The provision of a window with 3D avatars that render automatic translation of audiovisual content into sign languages enables a more significant offering of accessible programming, both on live television and video-on-demand services, while also reducing operational costs, particularly those associated with interpreters. However, additional challenges involve synchronizing the original audio with its equivalent presentation in signs, which is slower than human speech. Studies indicate that the duration of a sign presentation in American Sign Language (ASL) is approximately twice the length of a monosyllabic word. This study aims to investigate methods and strategies to improve the synchronization of signs with the associated video content. One approach considered is the application of automatic sentence summarization (or compression), which aims to produce a shorter paraphrase of a given input sentence, simplifying it to require fewer signs to convey the original content. Another applicable technique is called elastic adjustment, which involves modifying the playback duration of a media object. Elastic adjustment algorithms can adjust the speed of audio by up to 10%, either up or down, while maintaining acceptable audio perception quality. The main hypothesis is that the combination of such methods will allow the time required to interpret content into Brazilian Sign Language (Libras) to avoid desynchronization or loss of context, prioritizing the preservation of the most crucial information related to the content. 30 experiments with automatic summarization of journalistic content were conducted. In some cases, it was possible to display Brazilian Sign Language (Libras) signing using a 3D avatar within the duration of the news segment. Finally, an analysis of the quality of the automatic translation was carried out using the Bilingual Evaluation Understudy (BLEU), Metric for Evaluation of Translation with Explicit ORdering (METEOR) and the Translation Edit Rate (TER) algorithms.

Keywords: Sign language, Video stream, Automatic translation, Accessibility

Resumo

A provisão de uma janela com avatares 3D que renderizam a tradução automática de conteúdo audiovisual para línguas de sinais permite uma oferta mais significativa de programação acessível, tanto na televisão ao vivo quanto em serviços de vídeo sob demanda, além de reduzir os custos operacionais envolvidos, especialmente com intérpretes. No entanto, desafios adicionais envolvem a sincronização do áudio original com a apresentação equivalente em sinais, que é mais lenta do que a fala humana. Estudos indicam que a duração da apresentação de um sinal na Língua de Sinais Americana (ASL) é aproximadamente o dobro da duração de uma palavra monossilábica. Este trabalho visa investigar métodos e estratégias para melhorar a sincronização dos sinais com o conteúdo de vídeo associado. Uma das abordagens consideradas é a aplicação de sumarização automática de sentenças (ou compressão), que visa produzir uma paráfrase mais curta para uma dada sentença de entrada em um processo de simplificação para exigir menos sinais para transmitir o conteúdo original. Outra técnica aplicável é chamada de ajuste elástico, que consiste em modificar a duração de exibição de um objeto de mídia. Algoritmos de ajuste elástico podem ajustar a velocidade do áudio em até 10%, para cima ou para baixo, mantendo a qualidade da percepção do áudio dentro de limites aceitáveis. A hipótese principal é que a combinação de tais métodos permita que o tempo necessário para interpretar o conteúdo na Língua Brasileira de Sinais não cause uma dessincronização das informações ou fuga do contexto, priorizando preservar as informações mais cruciais relacionadas ao conteúdo. 30 experimentos com sumarização automática de matérias jornalísticas foram realizados. Em alguns, foi possível exibir a sinalização em Língua Brasileira de Sinais usando um avatar 3D dentro da duração da notícia. Ao final, foi realizada uma análise da qualidade da tradução automática com os algoritmos Bilingual Evaluation Understudy (BLEU), Metric for Evaluation of Translation with Explicit ORdering (METEOR) e Translation Edit Rate (TER).

Palavras-chave: Língua de Sinais, Fluxo de Vídeo, Tradução Automática, Avatar 3D, Acessibilidade.

Prefácio

Esta dissertação de mestrado foi submetida à Universidade Federal da Paraíba (UFPB) como requisito parcial para obtenção do grau de Mestre em Informática.

A dissertação foi desenvolvida no Programa de Pós-Graduação em Informática (PPGI), tendo como orientador o Prof. Dr. **Guido Lemos de Souza Filho**.

Agradecimentos

Agradeço aos meus pais, pelo apoio aos estudos.

A minha esposa, pelo incentivo, inspiração, amor e amizade.

Ao orientador, Guido Lemos, por todos os ensinamentos.

Aos meus colegas do Lavid, em especial a Derzu Omaia, Tiago Maritan, Raoni Kuleza e Rostand Oliveira, pelo incentivo e companheirismo.

A Capes, por recursos destinados ao programa ao longo dos anos.

A RNP, pelas bolsas de P&D e pelos recursos que viabilizaram as idas à eventos para apresentação de trabalhos acadêmicos e tecnológicos.

*“Acessibilidade é tornar o mundo
melhor para todos.”*

– Lucas Gonçalves

Sumário

Abstract	i
Resumo	ii
Prefácio	iii
Agradecimentos	iv
Alinhamento com a Linha de Pesquisa	viii
Lista de Tabelas	ix
Lista de Figuras	x
Lista de Abreviações	xi
1 Introdução	1
1.1 Motivação	2
1.2 Hipotése	2
1.3 Objetivo	3
1.3.1 Objetivos específicos	3
1.4 Organização do Trabalho	4
2 Referencial Teórico	5
2.1 Língua de Sinais	5
2.2 Sumarização de texto	5
2.2.1 Tipos de Sumarização	6
2.2.2 Aplicações de Sumarização de Texto na Tradução para Libras	6
2.3 Ajuste Elástico de Tempo de Exibição	6
2.4 Métricas de Avaliação de Tradução	7
2.4.1 Métrica METEOR	7
2.4.2 Métrica BLEU	8
2.4.3 Comparação entre METEOR e BLEU	9
2.4.4 Métrica TER	9
2.5 VLibras	10

3	Detalhamento das técnicas estudadas	12
3.1	Sumarização de texto	12
3.2	Ajuste Elástico	13
3.3	Combinação de Técnicas: Cenário <i>on-demand</i>	13
4	Trabalhos Relacionados	15
4.1	Revisão da literatura	15
4.1.1	Considerações	16
5	Metodologia e Experimentação	17
5.1	Estruturação dos experimentos	17
5.2	Definição dos experimentos	17
5.3	Planejamento dos experimentos	18
5.3.1	Seleção de Contexto	18
5.3.2	Formulação da Hipótese	18
5.4	Seleção das Variáveis	18
5.5	Seleção dos vídeos	18
5.6	Projeto do Experimento	19
5.7	Instrumentação	19
5.8	Método de avaliação	20
6	Execução do experimento e Análise dos Resultados	22
6.1	Descrição dos dados	23
6.2	Qualidade de tradução	25
6.2.1	Resultados Observados	26
7	Conclusões e Trabalhos Futuros	32
7.1	Trabalhos Futuros	32
	Referências	35
A	Scripts	38

Alinhamento com a Linha de Pesquisa

Linha de Pesquisa: Computação Distribuída

Este trabalho endereça um problema de acessibilidade, com vistas a possibilitar a inclusão automatizada aos surdos que consomem conteúdos audiovisuais digitais. Dados divulgados pelo IBGE em 2021 informam que cerca de 2,3 milhões de pessoas possuem algum grau de surdez no Brasil. Uma vez solucionado o problema, a distribuição de conteúdo para geração automatizada de acessibilidade permitirá que os Surdos possam consumir conteúdos audiovisuais com tradução para língua de sinais Brasileira.

Lista de Tabelas

6.1	Listagem dos principais dados do experimento	24
6.2	Comparação dos scores METEOR e BLEU para tradução com sumarização, com e sem stemming	27
6.3	Comparação dos scores METEOR e BLEU para tradução sem sumarização, com e sem stemming	28
6.4	Médias dos escores BLEU da tradução com sumarização, com e sem stemming.	29
6.5	Médias dos escores BLEU da tradução sem sumarização, com e sem stemming.	29
6.6	Médias dos escores METEOR da tradução sumarizada, com e sem stemming.	29
6.7	Médias dos escores METEOR da tradução sem sumarização, com e sem stemming.	29
6.8	Comparação de TER entre Glosas geradas após sumarização e glosas geradas sem sumarização	30
6.9	Comparação dos scores METEOR, BLEU e TER para tradução pelo serviço de transcrição e sumarização em fase experimental	31
6.10	Médias dos escores METEOR da tradução do serviço em fase experimental, com e sem stemming.	31
6.11	Médias dos escores bleu da tradução do serviço em fase experimental, com e sem stemming.	31

Lista de Figuras

2.1	Site do Governo Brasileiro - VLibras Suite	10
6.1	Implementação do teste t de student pareado	23
6.2	Duração com acessibilidade vs duração vídeo sem acessibilidade	25
7.1	Janela com avatar intérprete de Libras na TV	33
7.2	Possibilidade de ajuste de posição de exibição	34
7.3	Janela com avatar intérprete de Libras em um dispositivo considerado segunda tela.	34

Lista de Abreviações

Abreviação	Descrição
BLEU	Avaliação Bilíngue Assistente (Bilingual Evaluation Understudy)
FUNADE	Fundação Centro Integrado de Apoio à Pessoa com Deficiência
IBGE	Instituto Brasileiro de Geografia e Estatística
Libras	Língua Brasileira de Sinais
LLM	Modelos de Linguagem de Grande Escala (Large Language Models)
Mcom	Ministério das Telecomunicações
MDHC	Ministério dos Direitos Humanos e da Cidadania
METEOR	Métrica para Avaliação de Tradução com Ordenação Explícita (Metric for Evaluation)
MGISP	Ministério da Gestão e Inovação em Serviços Públicos
SNDPD	Secretaria Nacional dos Direitos da Pessoa com Deficiência
RNP	Rede Nacional de Ensino e Pesquisa
SGBTVD	Sistema Brasileiro de TV Digital
OMS	Organização Mundial de Saúde

Capítulo 1

Introdução

A televisão (TV) tem um papel importante na sociedade brasileira. Além de fornecer entretenimento individual e coletivo, ela também atua como elemento informativo, educacional e de interação social [18] [22]. Segundo dados divulgados pelo IBGE em 2021 [8], há 2,3 milhões de pessoas com algum grau de surdez no Brasil. Já a Organização Mundial de Saúde, cita que 1,5 bilhão de pessoas têm algum grau de deficiência auditiva (surdez) hoje no mundo e, em 2050, é esperado que 1 em cada 4 pessoas tenham deficiência auditiva [15]. Para essas pessoas, é necessário que a informação esteja disponível através das línguas de sinais ¹, que são a sua língua natural de comunicação.

Por outro lado, o suporte para línguas de sinais é raramente explorado nas tecnologias de fluxo de vídeo. Na TV, por exemplo, o suporte a línguas de sinais é, em geral, limitado a uma janela com um intérprete de língua de sinais, apresentada juntamente com o vídeo original do programa, gravado ao vivo. Essa solução possui altos custos operacionais para geração e produção (câmeras, estúdio, equipe, etc.) dos conteúdos, necessita de intérpretes humanos em tempo integral caso insira-se o idioma de sinais em conteúdo gerado, o que acaba restringindo seu uso a uma pequena parcela da programação. Em plataformas de transmissão de vídeos por demanda, não existe exploração de um recursos que permitam a inclusão de línguas de modalidade gestual-visual. Essas dificuldades resultam em uma grande barreira para a comunicação com outras pessoas, o acesso a informações, a aquisição de conhecimentos, consumo de entretenimento, dentre outros.

Em 2012 foi realizado um experimento com usuários surdos para avaliação do nível de compreensão de vídeos com legendas e com tradução para Língua de Sinais Brasileira (libras) [2]. O experimento foi feito com 20 usuários surdos na Fundação Centro Integrado de Apoio à Pessoa com Deficiência (FUNAD) do estado da Paraíba. A seleção se realizou por conveniência dos cursos de formação em libras da FUNAD e

¹As línguas de sinais são línguas naturais de modalidade gestual-visual, em que gestual significa o conjunto de elementos linguísticos manuais, corporais e faciais necessários para a articulação e significação visual-cultural do sinal.

os usuários foram divididos aleatoriamente em 2 grupos de 10 usuários. Em uma das etapas do experimento, os usuários foram convidados a assistir conteúdos audiovisuais com legenda e conteúdos audiovisuais com tradução para libras. Cada conteúdo foi apresentado duas vezes e, após a apresentação de cada um desses conteúdos, os usuários foram convidados a responder questões subjetivas relacionadas ao conteúdo apresentado. [2] mostra que, considerando todo o conteúdo, a taxa média de acertos para os usuários que assistiram vídeos com traduções para libras com uma solução proposta foi de 79,38% com um desvio padrão de 9,34%, enquanto a taxa média de acertos para os usuários que assistiram vídeos apenas com legendas foi de 25,63% com um desvio padrão de 19,86%. Isso demonstra o impacto de uma solução na comunicação com as pessoas surdas.

1.1 Motivação

Devida a necessidade de uma alternativa para mitigar os problemas citados, a disponibilização de uma opção de uso de avatares 3D renderizando a tradução automática de conteúdo audiovisual para línguas de sinais permitirá uma oferta maior de programação acessível, reduzindo os custos operacionais envolvidos, sobretudo com intérpretes humanos. Entretanto, há desafios envolvidos na sincronização temporal do conteúdo com a apresentação da sua tradução para língua de sinais, que é mais lenta do que a fala humana [4]. Este trabalho visa investigar métodos e estratégias para melhorar a sincronização dos sinais com o conteúdo de vídeo associado. Uma das abordagens consideradas é a aplicação de sumarização automática de sentenças (ou compressão), que visa produzir uma paráfrase mais curta para uma dada sentença de entrada em um processo de simplificação para consequentemente precisar de menos sinais na tradução para Libras. Outra técnica possível de ser aplicada é a chamada de ajuste elástico, que consiste em modificar a duração de exibição de um objeto de mídia. Algoritmos de ajuste elástico podem ajustar a duração de áudio e vídeo em até 10%, para cima ou para baixo, mantendo a qualidade da percepção do áudio dentro de limites aceitáveis [10] [1].

Do ponto de vista tecnológico, a motivação base é definir uma forma de resumir o conteúdo falado em vídeos para realização da tradução para glosas de libras e reprodução utilizando um player com avatar 3d. Do ponto de vista social, essa motivação visa reduzir a falta de acesso a informações pela falta de acessibilidade, através do uso dessa tecnologia.

1.2 Hipótese

As hipóteses formuladas nesta pesquisa são apresentadas logo a seguir e visam direcionar o processo investigativo, além de embasar as análises e interpretações dos resultados experimentais. Cada hipótese foi desenvolvida com base nas lacunas identificadas na literatura, nos objetivos propostos e nas metodologias aplicadas. Dessa

forma, as hipóteses desta pesquisa são as seguintes:

1. É possível prover acesso à informação pra comunidade Surda através da tradução automática para LIBRAS a partir da sumarização automática dos *closed-captions*.
2. A tradução será exibida dentro do espaço de tempo do conteúdo que está sendo traduzido.

Formalmente, essas hipóteses podem ser representadas da seguinte forma:

1. Hipótese nula H0: Não há diferença significativa na duração do vídeo de tradução feita utilizando a suíte do VLibras com sumarização automática para a realizada sem sumarização.
2. Hipótese alternativa H1: A aplicação de sumarização automática nas legendas de vídeos resulta em diferenças significativas na duração do vídeo de tradução feita utilizando a suíte do VLibras, em comparação com as traduções feitas sem a sumarização.

No capítulo 5 é apresentado o detalhamento do experimento para testar essas hipóteses, com planejamento, definição e execução para testar e avaliar a hipótese geral.

1.3 Objetivo

O objetivo do trabalho é sincronizar a tradução para libras com o conteúdo traduzido. Uma das técnicas a serem utilizadas é a compressão automática as mensagens a serem traduzidas para língua de sinais, extraídas de legendas (transcritas ou ocultas - do inglês, *closed caption*), para então realizar o processo de tradução para GLOSSAS de libras. Alternativamente, é possível utilizar uma técnica de ajuste elástico de mídia de exibição, técnica que permite expansão do tempo de exibição do conteúdo audiovisual, sem percepção na experiência de consumo da mídia audiovisual.

1.3.1 Objetivos específicos

1. Identificar e analisar as limitações das ferramentas atuais de tradução automática para Libras no que diz respeito à sincronização de mensagens de vídeos, avaliando o tempo de resposta e a qualidade da tradução.
2. Desenvolver e testar um modelo de sincronização que ajuste automaticamente a duração da tradução para Libras mantendo a precisão sem sacrificar a naturalidade.
3. Avaliar a qualidade da tradução, gerada pela suíte do VLibras após a sumarização automática, com traduções humanas, utilizando métricas de qualidade como BLEU, METEOR e TER.

1.4 Organização do Trabalho

Este trabalho faz um estudo acerca de estratégias que permitam sincronização de línguas naturais de modalidade gestual-visual com conteúdo audiovisual, para que os sinais sejam reproduzidos dentro do espaço de tempo do vídeo traduzido, sem comprometimento de compreensão da mensagem original.

No capítulo 2 é apresentado o referencial teórico, que discutirá os principais conceitos relacionados a este estudo, como Libras, compressão de sentenças, ajuste elástico e a ferramenta VLibras. No capítulo 3, é apresentado o detalhamento das técnicas estudadas. No capítulo 4, são apresentados trabalhos relacionados. No capítulo 5, é apresentada a metodologia e o processo de experimentação. No capítulo 6, são apresentados resultados. Por fim, no capítulo 7 são apresentadas as conclusões e propostas de trabalhos futuros.

Capítulo 2

Referencial Teórico

Neste capítulo, são apresentados alguns conceitos teóricos extraídos de uma revisão da literatura acerca da solução proposta.

2.1 Língua de Sinais

As línguas de sinais são de grande interesse não apenas para os linguistas, mas também para a comunidade surda em geral, pois, embora sejam sistemas de comunicação convencionais, sua transmissão física é muito diferente da transmissão das línguas faladas [17].

A língua de sinais, um sistema complexo de comunicação gestual-visual, desempenha um papel importante na facilitação da interação entre indivíduos surdos e a sociedade em geral. Essa forma de comunicação é essencial para promover a inclusão e garantir a acessibilidade linguística para pessoas surdas, permitindo-lhes participar amplamente das atividades sociais, educacionais e profissionais.

Existem diversas línguas de sinais ao redor do mundo, cada uma com sua própria gramática e estrutura linguística distintas. No contexto brasileiro, a língua de sinais utilizada é a Língua Brasileira de Sinais (Libras) [7].

A representação de uma língua de sinais em texto é chamada glosa. A glosa é uma forma de expressão textual, que segue as normas gramaticais da língua gestual, com o propósito de tornar a comunicação mais acessível. Ela tem como objetivo facilitar a compreensão tanto para os usuários fluentes de Libras quanto para intérpretes, além de possibilitar a tradução e a comunicação textual entre os falantes da língua de sinais. [7].

2.2 Sumarização de texto

A sumarização de texto é uma tarefa essencial no campo do Processamento de Linguagem Natural (PLN), cujo objetivo é reduzir automaticamente o conteúdo

textual, mantendo a essência e as informações mais relevantes. Com o crescimento exponencial de dados textuais, a necessidade de técnicas eficazes para condensar informações se tornou cada vez mais premente em áreas como jornalismo, educação, redes sociais e acessibilidade, como na tradução para Libras

2.2.1 Tipos de Sumarização

As técnicas de sumarização de texto são amplamente divididas em duas abordagens principais: sumarização extrativa e sumarização abstrativa.

- **Sumarização Extrativa:** A sumarização extrativa seleciona as sentenças mais importantes diretamente do texto original. Para isso, diversas técnicas podem ser aplicadas, desde métodos simples, como a seleção de sentenças com base na frequência de palavras, até modelos mais avançados baseados em aprendizado de máquina. Técnicas como TF-IDF (*Term Frequency-Inverse Document Frequency*), redes neurais, e análise de relevância têm sido amplamente utilizadas para sumarização extrativa. A vantagem dessa abordagem é a preservação da gramática e coerência das frases, já que as sentenças extraídas permanecem inalteradas.
- **Sumarização Abstrativa:** Diferentemente da extrativa, a sumarização abstrativa gera um resumo que não contém apenas frases ou trechos do texto original, mas sim uma reformulação, criando novas frases. Isso implica em uma compreensão mais profunda do conteúdo original e a criação de um resumo que, muitas vezes, é mais conciso e fluido. Essa abordagem é mais complexa, exigindo modelos baseados em redes neurais profundas, como as Redes Neurais Recorrentes (RNNs) e os Transformers, que podem gerar linguagem natural a partir de grandes corpora de dados.

2.2.2 Aplicações de Sumarização de Texto na Tradução para Libras

No contexto da tradução automática para Libras, a sumarização de texto desempenha um papel crucial. Como a língua de Sinais tem uma estrutura gramatical distinta da língua escrita, a sumarização de textos longos pode ser fundamental para que a tradução para Libras seja mais eficiente, preservando o significado das mensagens sem comprometer o tempo de sinalização. Pesquisas recentes exploram a integração de modelos de sumarização com sistemas de tradução para Libras, como uma forma de melhorar a sincronização e a acessibilidade em vídeos, especialmente aqueles voltados para o público surdo.

2.3 Ajuste Elástico de Tempo de Exibição

O ajuste elástico de tempo de exibição é uma técnica utilizada com o intuito de adaptar dinamicamente a duração de objetos de mídia [3].

Essa técnica é especialmente relevante em ambientes de comunicação audiovisual, nos quais a sincronização entre a fala e as legendas desempenha um papel crucial na compreensão do conteúdo. O ajuste elástico de tempo de exibição permite uma adaptação dinâmica e personalizada, considerando as características individuais do espectador, como habilidades de leitura e processamento cognitivo. Isso promove uma experiência de visualização mais inclusiva e acessível, atendendo às necessidades de diferentes públicos, incluindo aqueles com deficiências visuais ou cognitivas.

Em apresentações hipermídia, umas das principais tarefas coordenadas pelo orquestrador da apresentação é a sincronização entre os diversos objetos componentes, que pode ser obtida através do ajuste elástico do tempo de exibição de objetos. Esta técnica pode ser aplicada em tempo de apresentação, para prevenir qualquer descasamento temporal causado pelos ambientes de transmissão e de execução. É possível realizar compressão e expansão do tempo de exibição de hipermídias, com sincronização inter mídia e reconstrução do relógio de referência [1].

O ajuste elástico de tempo de exibição representa um campo em constante evolução, com aprimoramentos contínuos em algoritmos e técnicas de implementação. A integração de métodos baseados em inteligência artificial e aprendizado de máquina tem permitido refinamentos adicionais nessa área, possibilitando uma adaptação mais precisa e eficiente das legendas aos diferentes contextos de visualização. Esses avanços contribuem para a criação de sistemas de legendagem mais sofisticados e acessíveis, ampliando o alcance e o impacto da comunicação audiovisual em diversos setores da sociedade.

2.4 Métricas de Avaliação de Tradução

Para avaliar a qualidade das traduções automáticas geradas pelo sistema VLibras em comparação com as traduções profissionais, foram utilizadas duas métricas amplamente reconhecidas na área de tradução automática: **METEOR** (*Metric for Evaluation of Translation with Explicit Ordering*), **BLEU** (*Bilingual Evaluation Understudy*) e **TER** (*Translation Edit Rate*).

2.4.1 Métrica METEOR

O **METEOR** foi desenvolvido para avaliar a qualidade das traduções automáticas com uma sensibilidade semântica maior em comparação com o BLEU. Enquanto o BLEU se baseia unicamente na correspondência exata de palavras, o METEOR considera:

- **Sinônimos:** A correspondência de palavras sinônimas é considerada.
- **Stemming:** A métrica ignora variações morfológicas ao reduzir as palavras às suas raízes.

- **Reordenamento:** O METEOR permite variações na ordem das palavras, desde que o significado seja preservado.

Essas características fazem com que o METEOR seja mais adequado para capturar aspectos de fluência e adequação semântica nas traduções.

Como exemplo de cálculo com a métrica METEOR, considere a seguinte tradução de referência (profissional) e a tradução automática:

Tradução de Referência: "O rápido desenvolvimento da tecnologia está mudando o mercado."

Tradução Automática: "O avanço veloz da tecnologia muda o mercado."

O METEOR leva em conta que "rápido" e "veloz" são sinônimos e que "mudando" e "muda" têm a mesma raiz morfológica. Por esse motivo, essa tradução receberia uma pontuação alta, mesmo que haja diferenças lexicais e de ordem.

O score METEOR varia de 0 a 1, sendo 1 a tradução perfeita.

2.4.2 Métrica BLEU

A métrica **BLEU** é uma das mais usadas para avaliar traduções automáticas. O BLEU calcula a **precisão** ao comparar os **n-gramas** (grupos de n palavras consecutivas) da tradução gerada com os da tradução de referência, levando em consideração apenas correspondências exatas de palavras.

Além disso, o BLEU aplica uma **penalidade de brevidade** para evitar que traduções curtas demais obtenham pontuações altas indevidamente.

Exemplo de Cálculo BLEU:

Tradução de Referência: "O rápido desenvolvimento da tecnologia está mudando o mercado."

Tradução Automática: "Tecnologia está mudando rapidamente o mercado."

A métrica BLEU verificaria quantos **n-gramas** (unigramas, bigramas, trigramas, etc.) da tradução automática correspondem aos da tradução de referência. Neste caso, o BLEU daria uma pontuação mais baixa, pois não considera sinônimos ou variações na ordem.

A fórmula do BLEU é dada por:

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

Onde:

- p_n é a precisão de n-gramas.
- N é o número máximo de n-gramas considerados.
- BP é a penalidade de brevidade.

2.4.3 Comparação entre METEOR e BLEU

O **METEOR** tende a ser mais sensível à semântica e variações morfológicas, enquanto o **BLEU** é mais rigoroso em termos de correspondência literal de palavras. O BLEU é amplamente utilizado, mas é criticado por não capturar bem a qualidade semântica de traduções. O METEOR, por sua vez, é mais flexível e leva em conta correspondências parciais, sendo particularmente útil em línguas com grande variação lexical, como a Língua Brasileira de Sinais (Libras).

2.4.4 Métrica TER

A métrica TER (*Translation Edit Rate*) é uma métrica de avaliação amplamente usada na tradução automática para medir a quantidade de edições necessárias para transformar uma hipótese de tradução (tradução gerada automaticamente) na tradução de referência (tradução correta ou esperada). Ela quantifica a quantidade de esforço necessário para corrigir uma tradução automática, comparando-a com uma tradução de referência.

Como funciona: O TER calcula o número mínimo de edições necessárias para fazer a tradução gerada coincidir com a tradução de referência. As edições incluem:

- Inserções: Adicionar uma palavra que está faltando na hipótese.
- Deleções: Remover uma palavra que está na hipótese, mas não deveria estar.
- Substituições: Substituir uma palavra incorreta por outra correta.
- Reordenações: Reordenar palavras para fazer a frase ter a ordem correta.

Fórmula do TER:

$$\text{TER} = \frac{\text{Número de Edições}}{\text{Número Total de Palavras na Tradução de Referência}} \quad (2.1)$$

O valor resultante indica o número de edições por palavra necessária para corrigir a tradução. Quanto menor o valor de TER, melhor é a qualidade da tradução, pois menos edições são necessárias para que a tradução coincida com a referência.

Exemplo: Se uma hipótese de tradução contiver 100 palavras e forem necessárias 20 edições (substituições, inserções, deleções ou reordenações) para torná-la idêntica à referência, a pontuação TER será:

$$\text{TER} = \frac{20}{100} = 0.2 \quad (2.2)$$

2.5 VLibras

A Suíte VLibras é um conjunto de ferramentas de código aberto que traduzem automaticamente conteúdos digitais de língua portuguesa para Libras, tornando informações mais acessíveis para pessoas surdas em computadores, dispositivos móveis, TVs, plataformas Web, entre outros [6].



Figura 2.1: Site do Governo Brasileiro - VLibras Suite

Na Figura 2.1 é exibida a página oficial do Vlibras. O projeto é resultado de uma parceria entre o Ministério da Gestão e Inovação em Serviços Públicos (MGISP), por meio da Secretaria de Governo Digital (SGD), o Ministério dos Direitos Humanos e da Cidadania (MDHC), por meio da Secretaria Nacional dos Direitos da Pessoa com Deficiência (SNDPD), e a Universidade Federal da Paraíba (UFPB), através do Laboratório de Aplicações de Vídeo Digital (LAVID) [6].

Os principais componentes do VLibras são:

- VLibras-Plugin e VLibras-Widget: extensões de navegador web que permitem que textos selecionados em páginas Web sejam traduzidos automaticamente para Libras e reproduzidas através de um avatar 3D;

- VLibras-Móvel: aplicação cliente do VLibras para dispositivos móveis (compatível com os sistemas Android e iOS);
- VLibras-Desktop: ferramenta utilizada para traduzir automaticamente textos selecionados em programas executados em computadores pessoais para Libras;
- VLibras-Vídeo: um portal que permite tradução para Libras de trilhas de áudio e legendas associadas ao vídeo. (Atualmente só disponível para vídeos legendados e sob demanda, pois o tempo para renderizar o vídeo com a tradução é maior que o tempo do vídeo.) ;

Outra parte do VLibras são os serviços de retaguarda (ou *backend*), denominados VLibras-services, que realizam a tradução automática para os outros componentes (ou ferramentas) e armazenam animações 3D dos sinais em Libras que são utilizadas para renderizar os conteúdos acessíveis após a tradução. Atualmente, o dicionário de sinais tem aproximadamente 21 mil sinais em Libras, uma maiores bases de dados deste tipo no mundo [6].

Há também uma ferramenta colaborativa, denominada WikiLibras, que permite que voluntários participem do processo de construção e expansão do dicionário de sinais [13].

A Suíte VLibras é uma iniciativa de extrema importância para assegurar o acesso das pessoas com deficiência auditiva à comunicação e à informação no contexto brasileiro. Funcionando como uma ferramenta vital para a inclusão digital desses indivíduos, a suíte proporciona uma série de benefícios substanciais. Entre esses benefícios destacam-se o cumprimento da legislação em brasileira, ao garantir a usabilidade e a acessibilidade para todas as pessoas, independentemente das ferramentas utilizadas ou de suas limitações individuais.

Além disso, a Suíte VLibras contribui para o aumento do número de usuários da Internet e para a melhoria geral da qualidade do acesso web, proporcionando informações, conhecimentos e meios de comunicação e expressão de forma equitativa para todos. Essa iniciativa não apenas visa melhorar a qualidade de vida, a empregabilidade, a autonomia e a inclusão social das pessoas com deficiência, mas também tem um impacto social amplo, almejando solucionar definitivamente a questão da acessibilidade digital no país para esse segmento da população.

Capítulo 3

Detalhamento das técnicas estudadas

“Knowing how things work is the basis for appreciation, and is thus a source of civilized delight.”

– William Safire

Este trabalho foca na proposta de pesquisa em relação à barreira de acessibilidade ainda existente em fluxos de vídeo devido à uma falta de sincronia de conteúdo audiovisual e gestual-visual.

3.1 Sumarização de texto

O processo de sumarização de texto consiste em extrair as partes principais de um texto reduzindo a quantidade de palavras utilizada para transmitir tais partes. Um algoritmo de compressão de sentenças em português foi proposto em [14], utilizando métodos variados de aprendizado de máquina, com sumarização extrativa.

Foram utilizados dois conjuntos de dados, nos quais existem pares de sentenças com sentenças longas (originais) e suas respectivas versões reduzidas. A abordagem utilizada envolve considerar que uma sentença sr é uma possível redução de outra sentença sl quando pode-se produzir sr após deletar nenhum ou mais tokens de sl . Desta forma, é possível ter alguns pares com sentenças idênticas (sr é igual a sl) em conjuntos de dados. Os autores destacam a importância da inclusão desses casos para ter exemplos com sentenças que não devem ser compactadas (já que, no aprendizado de máquina, os exemplos negativos são tão importantes quanto os positivos).

A compressão de sentenças em português foi tratada utilizando exclusão como uma tarefa de classificação de aprendizado de máquina: dada uma sentença s com uma sequência de tokens (incluindo sinais de pontuação) t_1, t_2, \dots, t_n , o objetivo é

construir um classificador que responda "sim" ou "não" à seguinte pergunta para cada token $t \in s$: "devemos remover este token t de s ?". Assim, cada token t em uma determinada sentença s em nossos conjuntos de dados é considerado uma instância.

Os resultados mostrados por [14] indicam que a proposta se mostrou mais eficiente do que as abordagens existentes para o idioma português. Devido ao fato de não ser possível reproduzir o método proposto utilizando o repositório disponibilizado pelos autores e considerando os recentes avanços de arquiteturas de modelos de linguagem de grande escala (LLMs) utilizando sumarização abstrativa, escolhemos o chatGPT4.0 para sumarização de *cosed-captions*, os quais comumente estão disponíveis para todos os conteúdos audiovisuais e, após a sumarização, realizar tradução para língua de sinais, gerando as glosas de libras. [21] mostra que as sumarizações utilizando LLMs, apesar de terem grandes diferenças estilísticas, como quantidade de paráfrases, foram consideradas equivalentes a resumos escritos por humanos.

3.2 Ajuste Elástico

Um algoritmo de compressão e expansão de fluxos de áudio comprimido foi proposto por [10]. [1] Propôs uma técnica para compressão e expansão de fluxos de vídeo MPEG através da estratégia de inserção e descarte de quadros MPEG. O padrão utilizado nos trabalhos é o MPEG-2, que é descrito pelas especificações ISO/IEC 13818.

Para expansão de fluxos de vídeo, o algoritmo proposto por [1] realiza a replicação de quadros do fluxo de entrada conforme a ordem de apresentação das figuras, obtida através da análise do tipo de cada figura e do armazenamento das figuras I ou P que foram utilizadas como referência na codificação de outras figuras. O algoritmo avalia, então, o momento em que essas figuras serão apresentadas para que sua replicação seja inserida na ordem correta dentro do fluxo de saída.

Na compressão de áudio proposta por [10], os algoritmos podem ajustar a duração de áudio em até 10% mantendo a qualidade da percepção dentro de limites aceitáveis.

3.3 Combinação de Técnicas: Cenário *on-demand*

Dado que a língua de sinais é mais lenta do que a fala humana [4], é possível, em um cenário *on-demand*, realizar a combinação das estratégias apresentadas, com o propósito de manter um sincronismo entre o audiovisual e o gestual-visual, contando com a possibilidade de expansão do tempo de exibição de mídia. Aplicando a compressão de sentenças nas legendas do conteúdo audiovisual para extrair as informações de forma resumida e, a partir disso, gerar glosas para serem renderizadas utilizando o componente de síntese de sinais (*player*) e os avatares 3D do VLibras.

Essa solução pode ser utilizada ainda com um ajuste de velocidade de reprodução dos sinais pelo *player*, de forma a acelerar a velocidade dos gestos do avatar, dentro

de um limite aceitável para compreensão por parte dos surdos.

A princípio, é possível utilizar o ajuste elástico de tempo de exibição apenas com conteúdos *on-demand*, onde é possível passar mais tempo para desocupar o buffer sem perder conteúdo. Em um cenário de TV ao vivo, não seria possível realizar o ajuste elástico no momento da recepção com o propósito de exibir a mídia por mais tempo, devido a ocupação do buffer de vídeo.

Vale mencionar que atualmente não é possível estimar o tempo necessário para reprodução dos sinais de libras a partir das glosas utilizando avatares, pois o dicionário de animações do VLibras não contém a duração das animações. Esse recurso poderá ser explorado de forma dinâmica no futuro.

Capítulo 4

Trabalhos Relacionados

Neste capítulo, são apresentados alguns trabalhos relacionados a machine translate extraídos de uma revisão da literatura e de grupos de pesquisas Brasileiros.

4.1 Revisão da literatura

Visando identificar os principais trabalhos relacionados ao tema desta pesquisa, foi realizada uma pesquisa na literatura sobre tradução automática de texto para língua de sinais.

[9] trouxe uma revisão sistemática sobre tradução automática de texto para língua de sinais (método conhecido na literatura como *Sign Language Machine Translation*), destacando projetos como o ViSiCAST e o ProDeaf. Eles classificaram os sistemas de tradução em três categorias principais e apresentaram uma análise crítica das limitações dos métodos convencionais, além de destacar o uso de tecnologias mais recentes, como redes neurais profundas para melhorar a tradução automática. No total, sessenta e oito trabalhos foram analisados. Entretanto, nenhum dos trabalhos revisados no documento menciona a restrição de tempo para exibição sincronizada com vídeo usando avatar. A maior parte dos estudos foca em métricas de tradução, desempenho do avatar e qualidade dos sinais, mas sem menções a restrições temporais de exibição do conteúdo traduzido em relação a vídeo.

[16] propôs um dispositivo assistivo baseado em um framework para realizar a tradução em tempo real de fala para a língua de sinais a partir de vídeos de *streaming*. As principais contribuições incluem o processamento em tempo real, no sentido de buscar animações existentes no momento em que o texto chega e o processo de tradução é feito com identificação de tópicos para redução do espaço de busca de tradução. Além disso, os autores implementaram um sistema de Feedback para ser usado com Rastreamento Ocular. Os experimentos foram realizados com vídeos educativos extraídos de um canal do YOUTUBE, acumulando aproximadamente oito horas de conteúdo. Foram geradas as transcrições dos vídeos e na sequencia

as transcrições foram segmentadas em sentenças e frases. Em seguida, refinaram a lista manualmente para remover redundâncias, o que resultou em seiscentas e trinta e duas sentenças. Usando técnicas de generalização, a lista foi estendida para mil seiscentos e dezessete sentenças. Após isso, com a ajuda de especialistas em ASL, traduziram a lista para texto em ASL (glosas). Essas traduções foram utilizadas para efeitos de comparação, entretanto não foi dito se as traduções foram feitas com algum tipo de resumo visando respeitar as durações de exibição dos vídeos. Apesar de ter bons resultados nas avaliações dos experimentos, não há muitas informações relacionadas ao cumprimento da restrição de tempo total de exibição dos vídeos juntamente com a tradução para língua de sinais.

[19] propôs uma modelagem e síntese de expressões faciais de Libras para avatares tridimensionais sinalizantes desenvolvidos em estilo realista. Como resultado final, foi desenvolvida uma metodologia para a criação de animações faciais para serem incorporadas à avatares sinalizantes.

[11] trouxe experimentos e soluções para inserção de movimentos suaves e naturais em avatar de sinalização de língua de sinais. Na área de animação de avatares para Línguas de Sinais, o trabalho propôs um método para animação fluida entre sinais em línguas de sinais, visando melhorar a transição e a naturalidade dos movimentos no contexto de avatares. Esse método representa uma importante contribuição para a síntese e animação de sinais em avatares 3D.

4.1.1 Considerações

Diversas pesquisas foram realizadas na área de acessibilidade para surdos, abordando soluções para tradução automática, síntese de sinais com avatares e métodos baseados em regras, corpus e redes neurais. Contudo, nenhuma das abordagens estudadas apresentou uma solução prática para resolver a dissincronia entre a língua de sinais e o idioma falado, especialmente em cenários de tradução em tempo real para vídeos. Essa lacuna ressalta a necessidade de desenvolver métodos mais precisos e adaptáveis que integrem a sincronização temporal como um aspecto central, garantindo uma experiência mais natural e compreensível para o usuário final.

Capítulo 5

Metodologia e Experimentação

Esse capítulo descreve os experimentos que têm como objetivo avaliar a proposta da solução. [20] descreve como testar hipóteses com experimentos: Hipóteses são, em geral, testadas em ambientes controlados, manipulando uma variável por vez enquanto se mantém as outras fixas.

5.1 Estruturação dos experimentos

[20] explana os seguintes pontos de experimento: Descrição, Estratégia, Operação, Avaliação e interpretação dos resultados e, por ultimo, Apresentação de resultados.

Com isso, será possível organizar os principais elementos do experimento, visando concentrar-se no controle e nas variáveis relevantes para a obtenção dos resultados.

5.2 Definição dos experimentos

Esta seção apresenta uma definição de experimento que tem como objeto de estudar a viabilidade da solução proposta afim de analisar a usabilidade da tradução acessível gerada com foco na experiência dos usuários surdos em um ambiente de consumo de vídeo no contexto de surdos brasileiros em um ambiente controlado.

O presente estudo investiga técnicas para redução no tempo necessário para exibição de um determinado conteúdo em língua de sinais na tradução de conteúdos audiovisuais, utilizando sumarização textual com o objetivo de traduzir um resumo de forma que a duração da mídia audiovisual seja suficiente para exibição da tradução em libras.

Objeto de estudo: Durações dos vídeos e das traduções para libras, com e sem sumarização, de cada material selecionado.

Propósito: Avaliar a viabilidade de tradução automática respeitando a duração de cada conteúdo para promover acessibilidade em larga escala

e de baixo custo.

Perspectiva: Reduzir do tempo necessário para reprodução da tradução para líbras utilizando a suíte VLibras, a partir da sumarização do conteúdo através do uso de LLM.

Contexto: O material com teor informativo contínuo, isso é, com notificação de informações ao longo de todo o vídeo.

5.3 Planejamento dos experimentos

5.3.1 Seleção de Contexto

O material selecionado, isso é, os vídeos que farão parte do experimento, serão extraídos do canal do YouTube da TV UFPB, da modalidade de Shorts vídeos.

5.3.2 Formulação da Hipótese

A hipótese alternativa mencionada na introdução, informalmente, foi:

A aplicação de sumarização automática nas legendas de vídeos resulta em diferenças significativas na duração do vídeo de tradução feita utilizando a suíte do VLibras, em comparação com as traduções feitas sem a sumarização.

Formalizando a hipótese, definindo medidas para avaliação, temos:

1. Hipótese nula H_0 : Não há diferença significativa na duração do vídeo de tradução feita utilizando a suíte do VLibras entre as traduções feitas com sumarização automática e as feitas sem sumarização
2. Hipótese alternativa H_0 : A duração do vídeo de tradução para Libras com sumarização é significativamente menor que a duração do vídeo com tradução sem sumarização.

5.4 Seleção das Variáveis

A variável independente desse estudo é a utilização é a quantidade de informação a ser traduzida. Podemos assumir os valores: (CSP) com a sumarização proposta e (SSP) sem a sumarização proposta. As variáveis dependentes são as variáveis de resposta aos diferentes valores ou tratamentos assumidos pela variável independente. No presente estudo, as variáveis dependentes são o tempo do vídeo e a acessibilidade para Surdos ao assistir os vídeos com traduções para libras.

5.5 Seleção dos vídeos

Os vídeos desse experimento serão os últimos trinta shorts vídeos do canal da TV UFPB no YouTube, escolhidos por conveniência por serem vídeos informativos com

fala de maneira contínua durante todo o tempo do vídeo, com durações entre vinte e quarenta e cinco segundos.

5.6 Projeto do Experimento

Os princípios gerais de planejamento serão explorados da seguinte maneira:

Balanceamento: O experimento será balanceado, pois cada conteúdo receberá o mesmo tipo de sumarização.

Bloqueio: Nenhuma abordagem sistemática de bloqueio será aplicada.

Para comparar os dois tratamentos, o teste estatístico a ser aplicado será o Teste t de Student pareado [12].

5.7 Instrumentação

O experimento utilizará trinta vídeos curtos (shorts) do canal TV UFPB no YouTube. Os vídeos foram previamente transcritos e as transcrições foram submetidas a um processo de sumarização automática, utilizando o a LLM do ChatGPT4.0. Esse processo pode ser feito de de transcrição em transcrição, via página web da LLM, ou pode-se utilizar a API da LLM. Para não sofrer com restrição uso de recursos diários do chatGPT4.0, optou-se por assinar a versão Plus do ChatGPT. Além disso, considerando a quantidade de vídeos, optou-se por realizar as sumarizações caso a caso para acompanhamento do texto produzido pela sumarização. As sumarizações foram feitas seguindo regras definidas para substituição de nomes próprios, locais e sites, conforme descrito a seguir.

Substituição de nomes próprios: Nomes de pessoas foram substituídos por termos genéricos, como "profissionais", "treinador", ou "pesquisador", garantindo a neutralidade e sinais de libras correspondentes das informações, para evitar a datilologia - sinalização de letra por letra.

Substituição de localidades: Nomes de cidades e estados foram substituídos por descrições gerais, como "cidade brasileira", "estado americano", exceto em casos relacionados à UFPB, que foi mantida inalterada para preservar a relevância institucional e considerando que UFPB tem um sinal específico.

Substituição de sites e instagrans por "SITE FRASE" e "INSTAGRAM PRÓPRIO", respectivamente: Todos os links de sites foram substituídos pela expressão "SITE FRASE" ou "INSTAGRAM PRÓPRIO, exceto menções ao SIGAA, que foram mantidas como parte essencial para garantir a clareza no contexto educacional, já que se trata do sistema de gestão acadêmico da UFPB.

A instrumentação consistirá em duas fases principais. Primeiro, a fase de tradução com a suíte VLibras, onde as transcrições originais dos vídeos, assim como as

transcrições sumarizadas, serão inseridas na suíte VLibras. Na segunda fase, serão avaliados dois cenários:

Cenário 1 (Controle): Tradução das transcrições originais sem sumarização.

Cenário 2 (Experimental): Tradução das transcrições após a sumarização utilizando chatGPT4.0.

Cenário 3 (Experimental): Tradução das transcrições após a sumarização utilizando produto ¹ resultado de pesquisa realizada por colegas do laboratório de aplicações de vídeo digital, objetivando sumarização de vídeos para geração de glosas.

A suíte VLibras será utilizada para gerar traduções em Libras para ambos os cenários, utilizando a opção de reprodução do sinal em 2x, e os vídeos resultantes serão analisados quanto as suas durações. O uso da suíte VLibras é obrigatório, visto que ela oferece uma plataforma consolidada e já aplicada no contexto nacional, com aproximadamente 21 mil sinais no dicionário de animações.

Para o cenário 3, será utilizada credencial de acesso à uma API (Interface de Programação de Aplicação) de Vídeo sob demanda, que permite a comunicação entre diferentes sistemas de software, facilitando integrações de funcionalidades.

5.8 Método de avaliação

O experimento será conduzido localmente, utilizando computador de mesa para processar os vídeos e as traduções. Não há restrição quanto ao ambiente de execução, podendo ser realizado em ambientes de trabalho ou residenciais. Após a geração das traduções, os resultados serão avaliados nos seguintes termos:

- Duração dos vídeos traduzidos pela suíte VLibras (comparando as traduções com e sem sumarização).
- Qualidade das traduções, comparando o desempenho da suíte VLibras com as traduções de profissionais, utilizando as métricas METEOR, BLEU e TER.
- Qualidade das traduções, comparando o desempenho das glosas extraídas com a API de vídeo sob demanda e as traduções de profissionais.

Será realizada a comparação com traduções de profissionais de Libras: Para fins de comparação, as mesmas transcrições também foram traduzidas por profissionais de Libras, usando técnica própria para sumarização que visa respeitar o tempo total do vídeo. Essas traduções profissionais serão utilizadas para verificar a qualidade das traduções geradas pela suíte VLibras após a sumarização. O objetivo é verificar se a tradução com sumarização se aproxima das traduções profissionais, utilizando as métricas *Bilingual Evaluation Understudy* (BLEU), *Metric for Evaluation of Translation with Explicit ORdering* (METEOR) e *Translation Edit Rate* (TER).

¹Veja mais em: <https://api2.assista.dev.v4h.cloud/api/v1/docs>

Para análise da qualidade de tradução, será utilizada a biblioteca NLTK (*Natural Language Toolkit*), desenvolvida em Python, projetada para trabalhar com processamento de linguagem natural (NLP - *Natural Language Processing*). Ela fornece uma ampla gama de ferramentas, recursos e bibliotecas que permitem a manipulação de dados textuais, análise linguística e desenvolvimento de aplicações de NLP.

Desenvolvida inicialmente por Steven Bird e Edward Loper na Universidade da Pensilvânia, a NLTK foi concebida como uma ferramenta educacional e de pesquisa para ajudar tanto estudantes quanto pesquisadores na área de processamento de linguagem natural. A NLTK oferece recursos que facilitam a análise e o processamento de texto, como ferramentas para dividir texto em sentenças ou palavras, Reduz palavras às suas formas bases e Métricas para Avaliação de Traduções com Ferramentas para calcular métricas, como BLEU e METEOR.

Já para o cálculo do TER, será utilizada biblioteca SacreBLEU, já que a NLTK não dispõe da implementação da métrica TER.

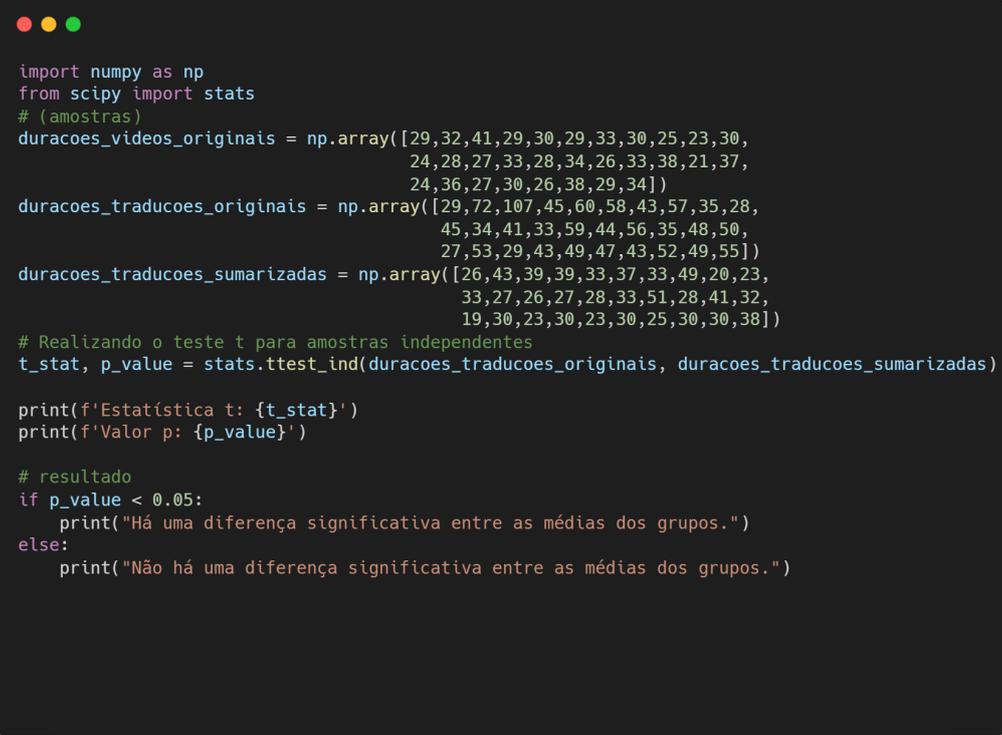
Por fim, esses dados quantitativos e qualitativos serão utilizados para responder às questões levantadas pelo experimento e analisar as hipóteses formuladas.

Capítulo 6

Execução do experimento e Análise dos Resultados

Neste capítulo, serão apresentados os experimentos de sumarização automática utilizando a LLM do chatGPT4.0 com o objetivo de avaliar a redução do tempo necessário para reprodução da acessibilidade junto aos conteúdos dos vídeos traduzidos. Os experimentos focaram em avaliar a capacidade do modelo de linguagem em produzir sumarizações, que preservam a integridade semântica e fluidez do conteúdo original, ao mesmo tempo que otimizam a quantidade de sinais necessários para a tradução em Língua Brasileira de Sinais (Libras), de forma que o tempo necessário para reprodução de sinais através de um *player* de libras se aproxime do tempo do conteúdo de vídeo traduzido. A motivação desse experimento é avaliar a hipótese levantada no Capítulo 1, juntamente com as análises estatísticas e interpretação dos resultados obtidos.

6.1 Descrição dos dados



```

import numpy as np
from scipy import stats
# (amostras)
duracoes_videos_originais = np.array([29,32,41,29,30,29,33,30,25,23,30,
                                       24,28,27,33,28,34,26,33,38,21,37,
                                       24,36,27,30,26,38,29,34])
duracoes_traducoes_originais = np.array([29,72,107,45,60,58,43,57,35,28,
                                          45,34,41,33,59,44,56,35,48,50,
                                          27,53,29,43,49,47,43,52,49,55])
duracoes_traducoes_sumarizadas = np.array([26,43,39,39,33,37,33,49,20,23,
                                           33,27,26,27,28,33,51,28,41,32,
                                           19,30,23,30,23,30,25,30,30,38])

# Realizando o teste t para amostras independentes
t_stat, p_value = stats.ttest_ind(duracoes_traducoes_originais, duracoes_traducoes_sumarizadas)

print(f'Estatística t: {t_stat}')
print(f'Valor p: {p_value}')

# resultado
if p_value < 0.05:
    print("Há uma diferença significativa entre as médias dos grupos.")
else:
    print("Não há uma diferença significativa entre as médias dos grupos.")

```

Figura 6.1: Implementação do teste t de student pareado

Para análise dos dados, foi aplicado um teste estatístico chamado t de Student pareado, descrito em [12], encontrado no módulo SciPy (*A scientific computing package for Python*), como pode ser visto na Figura 6.1. O teste foi realizado utilizando as durações dos vídeos de tradução de conteúdo, sem sumarização e com sumarização.

O teste t pareado compara a diferença entre as duas condições em cada par de observações e calcula se essa diferença média é estatisticamente significativa. A estatística t é calculada usando:

$$t = \frac{\bar{d}}{s_d/\sqrt{n}}$$

Onde \bar{d} é a média das diferenças entre os pares de observações, s_d é o desvio padrão das diferenças e n é o número de pares de observações. O valor de t é então comparado a uma distribuição t de Student para determinar o valor de p . O valor de p : O valor p associado à estatística t , que indica se a diferença é significativa.

Se o valor de p for menor que 0.05, a hipótese nula é rejeitada e podemos concluir que há uma diferença significativa entre as duas condições.

Se

$$p \geq 0.05$$

, não há evidências suficientes para afirmar que há uma diferença significativa.

vídeo	duração	VLibras(s)	Sumarizado VLibras(s)	Qtd Palavras Original	Qtd Palavras Sumarização
1	29s	29s	26s	43	39
2	32s	1m12s	43s	63	37
3	41s	1m47s	39s	105	58
4	29s	45s	39s	56	49
5	30s	1m	33s	63	39
6	29s	58s	37s	57	50
7	33s	43s	33s	61	46
8	30s	57s	49s	66	63
9	25s	35s	20s	48	37
10	23s	28s	23s	52	37
11	30s	45s	33s	56	45
12	24s	34s	27s	58	46
13	28s	41s	26s	60	30
14	27s	33s	27s	48	31
15	33s	59s	28s	70	43
16	28s	44s	33s	55	45
17	34s	56s	51s	70	65
18	26s	35s	28s	57	44
19	33s	48s	41s	61	55
20	38s	50s	32s	65	48
21	21s	27s	19s	44	29
22	37s	53s	30s	73	42
23	24s	29s	23s	49	41
24	36s	43s	30s	71	55
25	27s	49s	23s	56	37
26	30s	47s	30s	61	49
27	26s	43s	25s	56	44
28	38s	52s	30s	66	47
29	29s	49s	30s	57	48
30	34s	55s	38s	66	53

Tabela 6.1: Listagem dos principais dados do experimento

A análise feita nos indica um valor p 5.71113345309142e-06 e uma estatística t 4.995478781111571e-06.

Isso significa que há uma diferença significativa entre os dois grupos com tradução para libras. Na tabela 6.1 é possível visualizar como os dados se comportaram com a sumarização descrita na seção 5, enquanto na figura 6.2 é possível ter uma visão em relação duração dos vídeo do experimento e a duração do conteúdo de acessibilidade automática, com e sem sumarização de conteúdo.

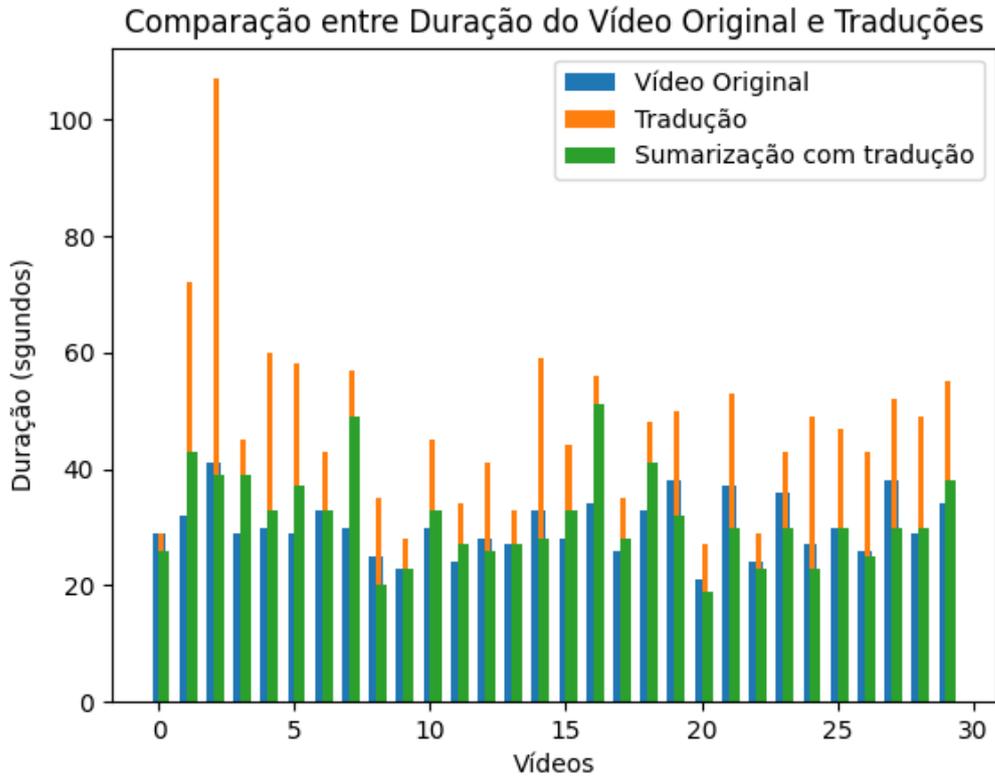


Figura 6.2: Duração com acessibilidade vs duração vídeo sem acessibilidade

É possível visualizar que apenas em um caso a duração do vídeo de tradução para libras foi idêntica a duração do conteúdo original. Nos demais casos, a duração do vídeo com acessibilidade sem sumarização foi maior do que o vídeo que estava sendo traduzido para libras. Entretanto, é possível notar que ainda não foi possível atingir o objetivo em todos os casos, apesar de se ter uma grande redução no tempo necessário para apresentação de acessibilidade dos conteúdos.

6.2 Qualidade de tradução

Durante o experimento, foram analisadas as métricas de qualidade de tradução tanto sobre as traduções de glosas sem a aplicação de *stemming* (que reduz as palavras

às suas raízes, ignorando variações morfológicas) quanto sobre as traduções após a aplicação de *stemming* .

6.2.1 Resultados Observados

A análise inicial com METEOR mostrou que as traduções realizadas pelo VLibras sobre o conteúdo sumarizado, quando comparadas com as traduções profissionais, apresentaram um score consistente, especialmente após a aplicação de *stemming*, o que sugere que, embora as palavras possam diferir em sua forma original, o VLibras consegue manter uma boa correspondência semântica.

O BLEU, por sua vez, apresentou um score ligeiramente inferior, uma vez que essa métrica penaliza mais fortemente pequenas variações nas palavras e na estrutura da sentença. No entanto, quando o *stemming* foi aplicado, o BLEU também mostrou um aumento nos scores, o que reforça a ideia de que as traduções automáticas do VLibras capturam a essência das glosas, mesmo que em alguns casos a forma exata das palavras varie. Os dados podem ser vistos na Tabela 6.2.

Entretanto, quando realizadas análises das métricas para as traduções sem sumarização tendo como referência as traduções produzidas pela intérprete, nota-se um pequeno aumento nos escores, como pode ser visto na Tabela 6.5. Isso se dá devido à escolha das informações a serem preservadas, tendo em vista que com a sumarização a mensagem possui menos palavras, ao passo que sem sumarização a disincronia é elevada.

Apesar disso, quando analisada a métrica TER com a biblioteca SacreBLEU, dos trinta casos, em vinte e sete são necessárias menos modificações na tradução da sumarização em relação a referência do que na tradução sem sumarização, como pode ser visto na Tabela 6.8.

METEOR Sem stemming	BLEU Sem stemming	METEOR Com stemming	BLEU Com stemming
0.558718	0.520541	0.558718	0.520541
0.476882	0.510103	0.476882	0.510103
0.243668	0.283896	0.231541	0.283896
0.261584	0.272078	0.261584	0.272078
0.283248	0.366667	0.430507	0.433333
0.234280	0.318979	0.248960	0.345561
0.546647	0.556286	0.568249	0.584101
0.307531	0.454328	0.401184	0.484617
0.492681	0.538278	0.541653	0.615175
0.461652	0.558002	0.461652	0.558002
0.402632	0.456550	0.455596	0.539560
0.346154	0.358485	0.438085	0.448106
0.620029	0.622235	0.620029	0.622235
0.220660	0.343927	0.256197	0.378320
0.302561	0.393631	0.410727	0.429416
0.519142	0.559690	0.541215	0.590784
0.362248	0.447368	0.376986	0.473684
0.451020	0.551561	0.472846	0.588331
0.571582	0.600000	0.613591	0.657143
0.490163	0.536769	0.508207	0.563608
0.425552	0.444491	0.425552	0.444491
0.250108	0.282359	0.284015	0.302528
0.481334	0.519561	0.556716	0.599493
0.375613	0.437543	0.427803	0.489019
0.630100	0.616551	0.630100	0.616551
0.485261	0.501530	0.547279	0.554323
0.548389	0.582592	0.548389	0.582592
0.367141	0.453639	0.367141	0.453639
0.424929	0.433403	0.424929	0.433403
0.441808	0.423241	0.505397	0.517294

Tabela 6.2: Comparação dos scores METEOR e BLEU para tradução com sumariação, com e sem stemming

Essa análise complementar usando as métricas METEOR e BLEU forneceu uma visão da performance do VLibras em relação às traduções profissionais. A aplicação do stemming foi um passo crucial para obter uma análise mais precisa, especialmente com a métrica BLEU, que é mais sensível a variações léxicas. Na Tabelas 6.4 e 6.6 podemos verificar as estatísticas das amostras sumarizadas pelo chat GPT4.0.

METEOR Sem stemming	BLEU Sem stemming	METEOR Com stemming	BLEU Com stemming
0.654324	0.700000	0.654324	0.700000
0.451192	0.444444	0.471730	0.472222
0.299122	0.344828	0.288124	0.344828
0.390261	0.386364	0.390261	0.386364
0.329922	0.342857	0.398919	0.371429
0.269090	0.348837	0.296832	0.372093
0.470443	0.500000	0.488775	0.526316
0.317631	0.421053	0.410908	0.447368
0.545070	0.512195	0.573980	0.585366
0.432558	0.454545	0.432558	0.454545
0.477933	0.461538	0.555743	0.538462
0.301544	0.361111	0.383717	0.416667
0.680693	0.666667	0.680693	0.666667
0.355859	0.378378	0.374497	0.378378
0.350067	0.451613	0.460766	0.451613
0.433194	0.468750	0.468750	0.500000
0.433594	0.439024	0.450490	0.463415
0.632716	0.620690	0.661408	0.655172
0.563322	0.525000	0.584245	0.550000
0.488767	0.500000	0.557553	0.547619
0.531120	0.555556	0.531120	0.555556
0.387663	0.446809	0.431796	0.489362
0.402450	0.413793	0.477965	0.482759
0.334860	0.450000	0.376643	0.525000
0.587918	0.527778	0.587918	0.527778
0.546706	0.567353	0.610192	0.621386
0.621568	0.558824	0.621568	0.588235
0.414508	0.414634	0.432236	0.439024
0.364583	0.388889	0.395494	0.416667
0.534442	0.534884	0.650086	0.651163

Tabela 6.3: Comparação dos scores METEOR e BLEU para tradução sem sumariação, com e sem stemming

Na Tabelas 6.5 e 6.7 podemos verificar as estatísticas das amostras sem sumarização.

Bleu	Sem Stemming	Com Stemming
Média	0.464809466666667	0.496397566666667
Mediana	0.455439	0.5136985
Variância	0.010104704281782	0.010593971135979

Tabela 6.4: Médias dos escores BLEU da tradução com sumarização, com e sem stemming.

Bleu	Sem Stemming	Com Stemming
Média	0.472880466666667	0.5041818
Mediana	0.453079	0.494681
Variância	0.008167915499916	0.00891302498076

Tabela 6.5: Médias dos escores BLEU da tradução sem sumarização, com e sem stemming.

Meteor	Sem Stemming	Com Stemming
Média	0.4194439	0.453057666666667
Mediana	0.43368	0.458624
Variância	0.013722769707823	0.012528876813822

Tabela 6.6: Médias dos escores METEOR da tradução sumarizada, com e sem stemming.

Meteor	Sem Stemming	Com Stemming
Média	0.453437333333333	0.489976366666667
Mediana	0.433394	0.47024
Variância	0.012826880973489	0.011619542244499

Tabela 6.7: Médias dos escores METEOR da tradução sem sumarização, com e sem stemming.

Após utilizar a API¹ de tradução de vídeo como serviço, para aproximadamente metade dos vídeos, foi possível extrair as glosas geradas através do token de cada mídia, solicitando os dados brutos presentes no servidor do serviço. Devido a indisponibilidade momentânea do serviço até a data de finalização desse trabalho, não foi possível analisar todos os 30 vídeos. Entretanto, foram realizados experimento

¹Disponível em: <https://api2.assista.dev.v4h.cloud/api/v1/docs/>

TER - Glosas com sumarização	TER - Glosas Sem Sumarização
64.29	71.43
58.62	100.0
80.85	104.26
82.86	105.71
103.57	114.29
81.08	108.11
65.71	102.86
84.85	87.88
71.79	74.36
80.0	116.0
75.0	66.67
81.82	93.94
56.67	80.0
89.66	103.45
74.07	92.59
71.88	90.63
73.68	76.32
70.37	59.26
76.47	94.12
63.89	94.44
65.38	69.23
84.09	90.91
76.0	104.0
71.05	92.11
70.0	106.67
72.97	70.27
72.41	79.31
84.85	103.03
72.22	91.67
69.05	76.19

Tabela 6.8: Comparação de TER entre Glosas geradas após sumarização e glosas geradas sem sumarização

com 16 Glosas geradas pelo serviço. É possível notar que a técnica proposta por esse trabalho tem um desempenho superior ao serviço experimental de tradução do grupo de pesquisa, dado que as métricas observadas nas Tabelas 6.9, 6.10 e 6.11 possuem índices de desempenho inferiores. Vale mencionar que as transcrições originais dos vídeos tiveram que ser retiradas para uso da api devido a uma indisponibilidade do serviço e limitações do trabalho acadêmico em curso que impediu o tratamento de arquivos de legendas. Cada vídeo teve uma transcrição realizada no Back-end do serviço e isso pode ter afetado o desempenho no comparativo.

METEOR Sem stemming	BLEU Sem stemming	METEOR Com stemming	BLEU Com stemming	Score TER
0.127273	0.171259	0.127273	0.171259	92.85714285714286
0.018450	0.014957	0.036900	0.029914	96.55172413793103
0.022779	0.018008	0.022779	0.018008	95.74468085106383
0.045593	0.047814	0.045593	0.047814	94.28571428571428
0.167669	0.131386	0.252800	0.183940	82.14285714285714
0.106383	0.111565	0.106383	0.111565	85.71428571428571
0.217807	0.226716	0.235443	0.226716	84.84848484848484
0.256148	0.134598	0.286434	0.148057	76.92307692307693
0.270249	0.183423	0.304260	0.209626	74.07407407407408
0.200178	0.096965	0.225870	0.113125	86.66666666666667
0.088908	0.044971	0.104789	0.056214	89.47368421052632
0.230533	0.121138	0.230533	0.121138	76.92307692307693
0.142130	0.094438	0.142130	0.094438	87.5
0.110630	0.021072	0.110630	0.021072	92.0
0.083799	0.094815	0.097765	0.110617	86.8421052631579
0.012255	0.005790	0.036765	0.017371	97.72727272727273

Tabela 6.9: Comparação dos scores METEOR, BLEU e TER para tradução pelo serviço de transcrição e sumarização em fase experimental

Meteor	Sem Stemming	Com Stemming
Média	0.131299	0.1478966875
Mediana	0.1189515	0.1189515
Variância	0.007303716664667	0.008930198059829

Tabela 6.10: Médias dos escores METEOR da tradução do serviço em fase experimental, com e sem stemming.

bleu	Sem Stemming	Com Stemming
Média	0.0949321875	0.105054625
Mediana	0.09589	0.111091
Variância	0.004354978925763	0.004821883721583

Tabela 6.11: Médias dos escores bleu da tradução do serviço em fase experimental, com e sem stemming.

Capítulo 7

Conclusões e Trabalhos Futuros

O presente estudo analisou técnicas para exibir conteúdo em vídeo com tradução automatizada para a língua de sinais. Foi possível concluir que métodos avançados de sumarização automática de texto podem melhorar significativamente a sincronização entre o conteúdo audiovisual e a interpretação em língua de sinais, aumentando assim a acessibilidade para a comunidade Surda. Os resultados preliminares são promissores, especialmente na capacidade de aproximar a duração dos vídeos em língua de sinais com o conteúdo original por meio de técnicas de sumarização. A integração de avatares 3D para tradução em língua de sinais apoia ainda mais esses esforços, oferecendo uma solução escalável para tornar o conteúdo multimídia mais inclusivo.

Apesar de nomes próprios terem sido substituídos por termos genéricos, nome de localidades também terem sido generalizados, sites e instagrans não terem sido detalhados nas sumarizações dos conteúdos (conforme regras estabelecidas no capítulo 5), é importante mencionar que quando há glosas que não estão presentes no dicionário de animações de libras do VLibras, o avatar realiza o processo de datilologia, o que leva mais tempo para sinalização, apesar de terem sido poucos os casos em que isso ocorreu.

Outro ponto a ser levado em consideração é que tradutores profissionais realizam resumos por tópicos com o objetivo sempre sinalizar em libras um conteúdo relacionado ao que está sendo falado no instante em que se está realizando a tradução. O trabalho atual se preocupou com o resumo do conteúdo inteiro, através da simplificação com redução de palavras do contexto, o que pode ter impacto nas métricas de avaliação da tradução.

7.1 Trabalhos Futuros

Como trabalho futuro, pretende-se realizar o aperfeiçoamento da técnica com o objetivo de atingir a sincronização total, com cálculo de estimativa de tempo necessário

para sinalização do conteúdo traduzido, para aplicação de técnica de ajuste elástico de forma dinâmica. Também serão realizados estudos com a comunidade Surda para avaliar a eficácia da sumarização automática em relação a compreensão do conteúdo e sua aceitação. Pesquisas futuras se concentrarão na otimização dos algoritmos de sumarização e na exploração de novas abordagens, como modelos de tradução automática neural, parametrização para avaliação de re-sumarização automática, considerando também a possibilidade de realizar mudança de velocidade na interpretação para a língua de sinais, dentro de um limite aceitável.

Há também uma aplicação de prova de conceito sendo desenvolvida para o novo padrão do Sistema Brasileiro de TV Digital Terrestre (SBTVD) [5].

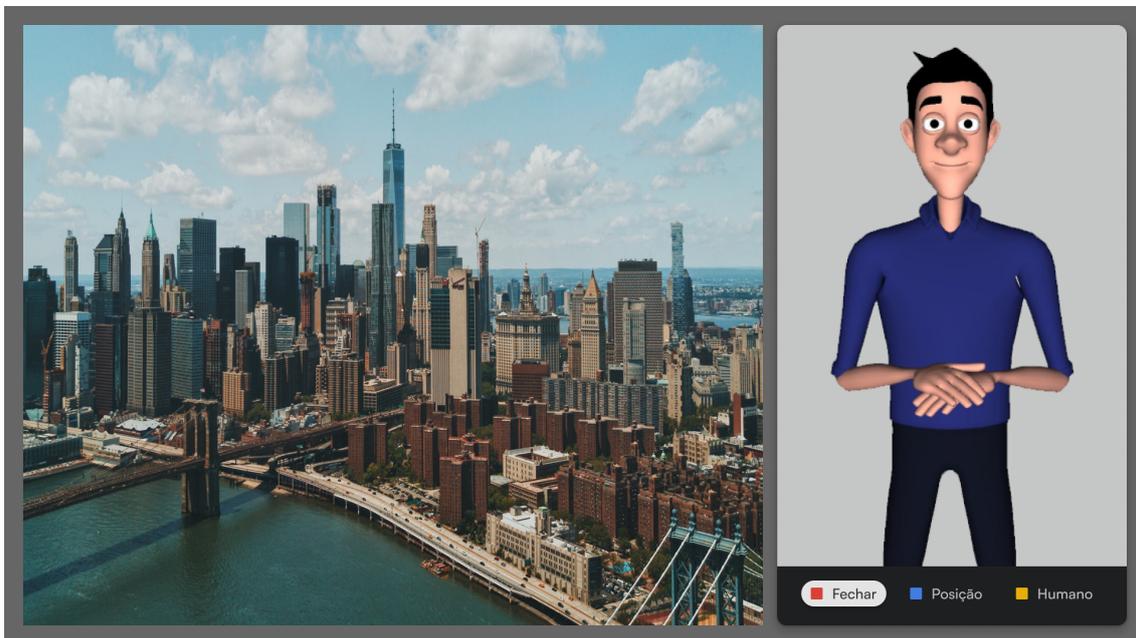


Figura 7.1: Janela com avatar intérprete de Libras na TV

O trabalho endereça requisitos de acessibilidade que estão sendo estudados e desenvolvidos, mais especificamente, o suporte a legendas ocultas, o suporte a língua de sinais, e o suporte a áudio descrição.

O objetivo é aplicar a solução no contexto de tv digital 3.0. Na solução, o radiodifusor poderá transmitir um fluxo contendo uma sequência de glosas, gerada e revisada por tradutores profissionais, que será convertido para Língua Brasileira de Sinais (Libras) na própria TV ou em um dispositivo de segunda tela, utilizando um *player* e avatares 3D. A proposta pode ser vista nas Figuras 7.1, 7.2 e 7.3.

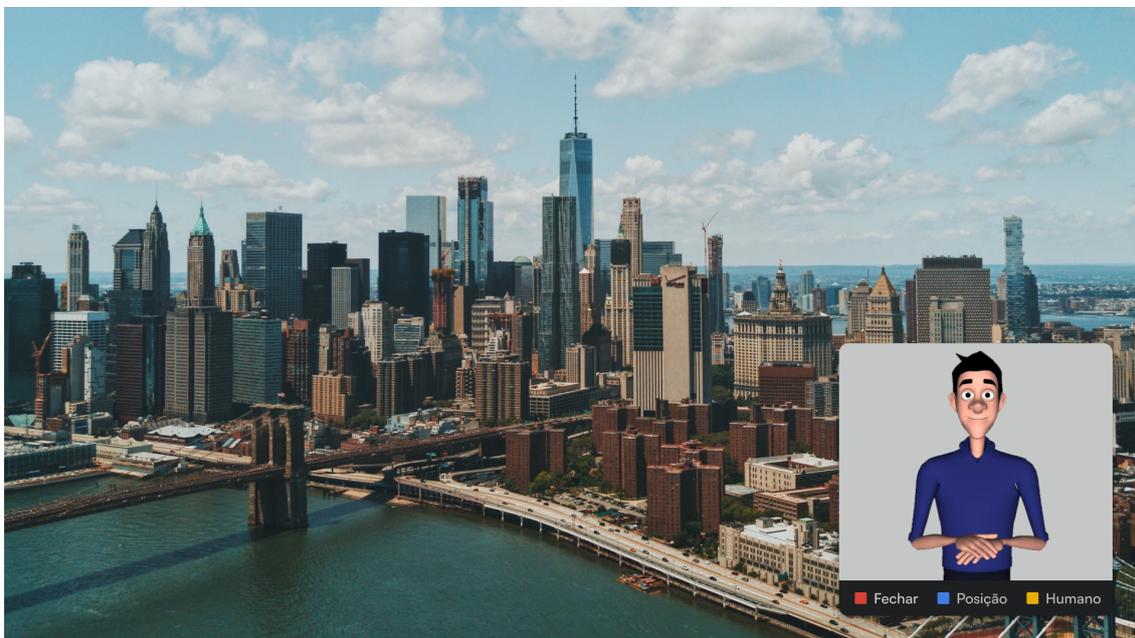


Figura 7.2: Possibilidade de ajuste de posição de exibição



Figura 7.3: Janela com avatar intérprete de Libras em um dispositivo considerado segunda tela.

Referências

- [1] Sergio Alves Cavendish. *ALGORITMO DE AJUSTE ELÁSTICO PARA VÍDEO EM FLUXOS MPEG-2*. PhD thesis, 2015.
- [2] Tiago Araújo. *Uma Solução Para Geração Automática De Trilhas Em Língua Brasileira De Sinais Em Conteúdos Multimídia*. PhD thesis, 2012.
- [3] Bruno Bachelet, Philippe Mahey, Rogério Rodrigues, and Luiz Fernando Soares. Elastic time computation in qos-driven hypermedia presentations. *Multimedia systems*, 12(6):461–478, 2007.
- [4] Ursula Bellugi and Susan Fischer. A comparison of sign language and spoken language. *Cognition*, 1(2-3):173–200, 1972.
- [5] Richelieu Costa, Derzu Omaia, Tiago Araújo, Jóison Pereira, Anderson Coutinho, Miguel Cruz, Victoria Pontes, Matheus Barbaosa, Abner Silva, and Guido L. S. Filho. Acessibilidade na tv 3.0 brasileira a partir de mídias de legenda, glosa e áudio descrição. In *Anais Estendidos do XXIX Simpósio Brasileiro de Sistemas Multimídia e Web*, pages 123–129, Porto Alegre, RS, Brasil, 2023. SBC.
- [6] Universidade Federal da Paraíba (UFPB). VLibras - Governo Digital. <https://vlibras.gov.br/>, 2023. Online; Accessed on August 30, 2023.
- [7] Maria Cecília Rafael de Góes. *Linguagem, surdez e educação*. Autores Associados, 2012.
- [8] IBGE. Pesquisa nacional de saúde 2019: País tem 17,3 milhões de pessoas com algum tipo de deficiência. <https://agenciadenoticias.ibge.gov.br/agencia-sala-de-imprensa/2013-agencia-de-noticias/releases/31445-pns-2019-pais-tem-17-3-milhoes-de-pessoas-com-algum-tipo-de-deficiencia>, 2021.
- [9] Navroz Kaur Kahlon and Williamjeet Singh. Machine translation from text to sign language: a systematic review. *Universal Access in the Information Society*, 22(1):1–35, jul 2021.
- [10] S. M. D. B. MARANHAO. *AJUSTE ELÁSTICO EM TEMPO DE EXIBIÇÃO PARA FLUXOS DE ÁUDIO COMPRIMIDO*. PhD thesis, 2006.

- [11] Leandro Martin. Método para animação entre sinais em línguas de sinais. Master's thesis, Universidade Estadual de Campinas, Campinas, SP, Brasil, 2021. Dissertação de Mestrado.
- [12] Douglas C. Montgomery and George C. Runger. *Applied Statistics and Probability for Engineers*. Wiley, 7th edition, 2018.
- [13] Danilo Assis Nobre, Mateus Ferreira, Tiago Maritan U de Araújo, Iris Regina Nascimento, Pollyane Carvalho, and Guido Lemos Filho. Wikilibras: Collaborative construction of a multimedia dictionary for brazilian sign language. In *Proceedings of the 17th Brazilian Symposium on Multimedia and the Web on Brazilian Symposium on Multimedia and the Web-Volume 1*, pages 244–251, 2011.
- [14] Fernando A. A. Nóbrega, Alipio M. Jorge, Pavel Brazdil, and Thiago A. S. Pardo. Sentence compression for portuguese. In Paulo Quaresma, Renata Vieira, Sandra Aluísio, Helena Moniz, Fernando Batista, and Teresa Gonçalves, editors, *Computational Processing of the Portuguese Language*, pages 270–280, Cham, 2020. Springer International Publishing.
- [15] OMS. Who: 1 in 4 people projected to have hearing problems by 2050. <https://www.who.int/news/item/02-03-2021-who-1-in-4-people-projected-to-have-hearing-problems-by-2050>, 2021.
- [16] Mwaffaq Otoom and Mohammad A. Alzubaidi. Ambient intelligence framework for real-time speech-to-sign translation. *Assistive Technology*, 30(3):119–132, 2018. PMID: 28152342.
- [17] Wendy Sandler and Diane Lillo-Martin. *Sign Language and Linguistic Universals*. Cambridge University Press, 2006.
- [18] Tatiana Aires Tavares, Celso Alberto Saibel Santos, Thiago Rocha de Assis, Clarissa Braga Bittencourt de Pinho, Germano Mariniello de Carvalho, and Clarissa Santana da Costa. A tv digital interativa como ferramenta de apoio à educação infantil. *Revista Brasileira de Informática na Educação*, 15(2):31–44, 2007.
- [19] Ackley Dias Will. *Modelagem e Síntese de Aspectos Faciais da Língua de Sinais Brasileira para Avatares Sinalizantes*. PhD thesis, Universidade Estadual de Campinas, Campinas, SP, Brasil, 2021. Tese de Doutorado.
- [20] Claes Wohlin, Per Runeson, Martin Höst, Magnus C. Ohlsson, Björn Regnell, and Anders Wesslén. *Experimentation in Software Engineering*. Springer Berlin Heidelberg, 2012.

-
- [21] Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B. Hashimoto. Benchmarking Large Language Models for News Summarization. *Transactions of the Association for Computational Linguistics*, 12:39–57, 01 2024.
- [22] Ângelo Piovesan. Vídeo e tv na educação. *Comunicação & Educação*, 1:105–112, 1994.

Apêndice A

Scripts

Nesta apêndice A encontram-se scripts de tradução de português para glosas de libras e um script de reprodução simultânea de múltiplos vídeos, para fins de comparação das traduções no contexto no vídeo traduzido.

Script para tradução de português para glosas:

```
1
2 from bs4 import BeautifulSoup
3 import argparse
4 import re
5 import os
6 import sys
7 import shutil
8 import requests
9 import json
10
11 def copy_file(src, dest):
12     """
13     Copia um arquivo de um diretório para outro.
14
15     Args:
16     src (str): Caminho completo do arquivo de origem.
17     dest (str): Caminho completo do diretório de destino.
18
19     Returns:
20     bool: True se o arquivo foi copiado com sucesso, False se ocorreu
21     ↪ um erro.
22     """
23     try:
24         # Verifica se o diretório de destino existe, senão o cria
25         if not os.path.exists(dest):
26             create_folder(dest)
```

```
26
27     # Caminho completo do arquivo de destino
28     dest_file_path = os.path.join(dest, os.path.basename(src))
29
30     # Copia o arquivo
31     shutil.copy(src, dest_file_path)
32
33     print(f'Arquivo copiado de "{src}" para "{dest_file_path}".')
34     return True
35 except Exception as e:
36     print(f'Ocorreu um erro ao copiar o arquivo: {e}')
37     return False
38
39 # Exemplo de uso:
40 # src_file_path = "caminho/do/arquivo/origem.txt"
41 # dest_directory_path = "caminho/do/diretorio/destino"
42
43 # copy_file(src_file_path, dest_directory_path)
44
45 def create_folder(folder_path):
46     """
47     Cria uma pasta no caminho especificado.
48
49     Args:
50         folder_path (str): O caminho da pasta a ser criada.
51
52     Returns:
53         bool: True se a pasta foi criada com sucesso, False se a pasta já
54             ↳ existe.
55     """
56     try:
57         os.makedirs(folder_path)
58         print(f' Pasta criada: {folder_path}')
59         return True
60     except FileExistsError:
61         print(f'A pasta já existe: {folder_path}')
62         return False
63     except Exception as e:
64         print(f'Ocorreu um erro ao criar a pasta: {e}')
65         return False
66
67 def extract_text_between_span_tags(text):
68     # Regex pattern to match text between <span> and </span> tags
69     pattern = r'<span>(.*?)</span>'
70     # Find all matches
71     matches = re.findall(pattern, text)
```

```
71     # Return the first match if found, else return an empty string
72     return matches
73
74 def write_lines_to_the_same_file(lines, folder_path='234',
75     ↪ filename='output', extension='.txt', encoding='utf-8'):
76
77     print('_texto'+filename+extension)
78
79     file_path = os.path.join(folder_path, 'ALL_'+filename+extension)
80     with open(file_path, 'a', encoding=encoding) as file:
81
82         # extract_text_between_span_tags(text)
83         # escrevendo texto sem tags
84
85
86
87     for line in lines:
88         file.write(extract_text_between_span_tags(line)[0]+ '\n')
89
90
91
92
93     # print(f"Conteúdo gravado em {filename} com codificação {encoding}")
94
95
96 def make_post_request(text):
97     url = 'https://traducao2.vlibras.gov.br/translate'
98     headers = {
99         'accept': 'text/plain',
100        'Content-Type': 'application/json'
101    }
102    data = {
103        "text": text
104    }
105
106    response = requests.post(url, headers=headers, data=json.dumps(data))
107
108    if response.status_code == 200:
109        print("Response:", response.text)
110        return response.text
111    else:
112        print(f"Failed to send request. Status code:
113        ↪ {response.status_code}")
114
```

```
115 def convert_time_format(time_str):
116     """
117     Converte uma string de tempo para o formato 'hh:mm:ss.sss'.
118
119     :param time_str: A string de tempo no formato 'hh:mm:ss.sss',
120     ↪ 'mm:ss.sss', 'ss.sss', ou 'ss.sss'.
121     :return: A string de tempo no formato 'hh:mm:ss.sss'.
122     """
123     # Remover o "s" do final, se presente
124     if time_str.endswith("s"):
125         time_str = time_str[:-1]
126
127     parts = time_str.split(':')
128     if len(parts) == 1: # Apenas segundos
129         seconds = float(parts[0])
130         hours = int(seconds // 3600)
131         minutes = int((seconds % 3600) // 60)
132         seconds = seconds % 60
133         return f"{hours:02}:{minutes:02}:{seconds:06.3f}"
134     elif len(parts) == 2: # Minutos e segundos
135         minutes = int(parts[0])
136         seconds = float(parts[1])
137         hours = minutes // 60
138         minutes = minutes % 60
139         return f"{hours:02}:{minutes:02}:{seconds:06.3f}"
140     elif len(parts) == 3: # Horas, minutos e segundos
141         hours = int(parts[0])
142         minutes = int(parts[1])
143         seconds = float(parts[2])
144         return f"{hours:02}:{minutes:02}:{seconds:06.3f}"
145     else:
146         raise ValueError("Formato de tempo inválido")
147
148 def parse_time_to_seconds(time_str):
149     """
150     Converte uma string de tempo no formato 'hh:mm:ss.sss', 'mm:ss.sss',
151     ↪ 'ss.sss', ou 'ss.sss' com ou sem zeros à esquerda para segundos.
152     """
153     # Remover 's' no final, se houver
154     if time_str.endswith('s'):
155         time_str = time_str[:-1]
156
157     # Regex para encontrar os padrões de tempo
158     pattern = re.compile(r'(?:(\d+):)?(?:\d+)?(\d+\.\d+|\d+)')
```

```
159     if not match:
160         raise ValueError(f"Formato de tempo inválido: {time_str}")
161
162     groups = match.groups()
163
164     # Inicializar horas, minutos e segundos
165     hours = 0
166     minutes = 0
167     seconds = 0
168
169     # Verificar a quantidade de grupos capturados e atribuir corretamente
170     if groups[2]: # Sempre há um grupo para os segundos
171         seconds = float(groups[2])
172     if groups[1]: # O grupo do meio pode ser minutos ou segundos
173         minutes = int(groups[1])
174     if groups[0]: # O primeiro grupo pode ser horas ou minutos
175         if groups[1]: # Se há grupo do meio, o primeiro é horas
176             hours = int(groups[0])
177         else: # Se não há grupo do meio, o primeiro é minutos
178             minutes = int(groups[0])
179
180     total_seconds = hours * 3600 + minutes * 60 + seconds
181     return total_seconds
182
183 def write_to_file(content, filename='output.txt'):
184     """
185     Grava o conteúdo fornecido em um arquivo.
186
187     :param content: O conteúdo a ser gravado no arquivo.
188     :param filename: O nome do arquivo. Padrão é 'output.txt'.
189     """
190     with open(filename, 'w') as file:
191         file.write(str(content))
192     print(f"Conteúdo gravado em {filename}")
193
194     # Exemplo de uso:
195     # content = "Este é um exemplo de texto a ser gravado no arquivo."
196     # write_to_file(content, 'meu_arquivo.txt')
197
198 def write_lines_to_file(lines, filename='output.txt', encoding='utf-8'):
199     """
200     Grava várias linhas fornecidas em um arquivo com a codificação
201     ↪ especificada.
202
203     :param lines: Uma lista de strings, cada uma representando uma linha
204     ↪ a ser gravada no arquivo.
```

```

203      :param filename: O nome do arquivo. Padrão é 'output.txt'.
204      :param encoding: A codificação a ser usada para gravar o arquivo.
205      ↪ Padrão é 'utf-8'.
206      """
207
208      # Cabeçalho XML
209      header = """<?xml version="1.0" encoding="UTF-8"?>
210      <tt
211      xmlns="http://www.w3.org/ns/ttml"
212      ↪ xmlns:tts="http://www.w3.org/ns/ttml#styling"
213      xmlns:ttp="http://www.w3.org/ns/ttml#parameter" xml:lang="en"
214      ↪ ttp:timeBase="media" ttp:frameRate="60"
215      ttp:frameRateMultiplier="1000 1001"
216      ttp:profile="http://www.w3.org/ns/ttml/profile/imsc1/image">
217      <head>
218      <styling>
219      <style xml:id="defaultStyle"
220      ↪ tts:fontFamily="proportionalSansSerif" tts:textOutline="black
221      ↪ 2%" tts:fontSize="100%" tts:textAlign="center"/>
222      <style xml:id="alignStart" tts:textAlign="start"
223      ↪ tts:color="yellow"/>
224      <style xml:id="alignCenter" tts:textAlign="center"
225      ↪ tts:color="yellow"/>
226      <style xml:id="alignEnd" tts:textAlign="end"/>
227      <style xml:id="narrador" tts:textAlign="start" tts:color="yellow"
228      ↪ tts:fontStyle="italic"/>
229      <style xml:id="lavid" tts:backgroundColor="white"/>
230
231      </styling>
232      <layout>
233      <region xml:id="rTop" tts:origin="5% 5%" tts:extent="90% 90%"
234      ↪ tts:displayAlign="before"/>
235      <region xml:id="rCenter" tts:origin="5% 5%" tts:extent="90% 90%"
236      ↪ tts:displayAlign="center"/>
237      <region xml:id="rBottom" tts:origin="5% 5%" tts:extent="90% 90%"
238      ↪ tts:displayAlign="after"/>
239      <region xml:id="area1" tts:origin="87.05% 0%" tts:extent="12.85%
240      ↪ 12.85%" tts:displayAlign="center"/>
241      </layout>
242      </head>
243      <body style="defaultStyle">
244      <div>
245      """
246
247      # Rodapé XML
248      footer = """ </div>

```

```
237     </body>
238 </tt>
239 """
240
241     with open(filename, 'w', encoding=encoding) as file:
242         # Escreve o cabeçalho
243         file.write(header)
244
245         # Escreve as linhas fornecidas
246         for line in lines:
247             file.write(line + '\n')
248
249         # Escreve o rodapé
250         file.write(footer)
251
252     print(f"Conteúdo gravado em {filename} com codificação {encoding}")
253
254 def main():
255     parser = argparse.ArgumentParser(description='Processa um argumento
256     ↪ de linha de comando.')
257     parser.add_argument('input_string', type=str, help='A string a ser
258     ↪ processada')
259
260     # Analisar argumentos da linha de comando
261     args = parser.parse_args()
262
263     # Acessar o argumento
264     input_string = args.input_string
265
266     # Exibir o argumento
267     print(f'Argumento recebido: {input_string}')
268
269     copy_file(input_string,
270     ↪ os.path.basename(input_string).split('.')[0]+'_glosas')
271
272     with open(input_string) as fp:
273         soup = BeautifulSoup(fp, 'xml')
274
275     #print(soup.prettify())
276
277     a = 0
278
279     for x in soup.find_all('p'):
280         #a+=1
281         #print(a)
282         x['begin']=convert_time_format(x.get('begin'))
```

```

280     x['end']=convert_time_format(x.get('end'))
281     span_tag = soup.new_tag('span') # Criar uma nova tag <span>
282     aux=make_post_request(str(x.string))
283     # print(aux)
284     span_tag.string =str(aux) # Definir o conteúdo da tag <span>
    ↪ como o conteúdo da tag <p>
285
286
287     # print(span_tag)
288     x.string = ""
289     x.append(span_tag) # Inserir a tag <span> antes da tag <p>
290     # print(x)
291     #exit()
292
293 a = 0
294
295 # print(soup)
296 formatted_content = soup.prettify()
297 with
    ↪ open(os.path.basename(input_string).split('.')[0]+'_glosas/ALL_glosa.ttml',
    ↪ 'w', encoding='utf-8') as f:
298     f.write(formatted_content)
299
300
301 # print(' vou mostrar o tamanho: ')
302 lista=soup.find_all('p')
303 # print( len(lista) )
304
305 #for x in range(len(lista)):
306 while len(lista):
307     lines=[]
308
309     for link in soup.find_all('p'):
310         # if(a>65):
311             #print(str(link.get('span')))
312             begin=parse_time_to_seconds(link.get('begin'))
313             if begin <= a + 2 :
314                 end=parse_time_to_seconds(link.get('end'))
315                 if end >= a:
316                     link['region']='rBottom'
317                     link['style']='alignCenter'
318                     print(str(a) + 'x' + str(begin) + 'x' + str(end) +
    ↪ str(link))
319                     lines.append(str(link))
320                     link.decompose()
321                     # write_to_file(str(link), str(str(a-2)+'.ttml'))

```

```

322         #elif end < a:
323             #link.extract();
324             #print('eu nem entrei aqui, rapaz')
325         #else:
326
327         lista=soup.find_all('p')
328         #print('tamanho da lista:' + str(len(lista)))
329         a+=2
330         if(not(len(lista)==0)): # & len(lines)== 0):
331             #lines.append(str(link))
332             write_lines_to_the_same_file(lines,
333                 ↪ os.path.basename(input_string).split('.')[0]+'_glosas',
334                 ↪ str('glosa'), '.txt', encoding='utf-8')
335
336             write_lines_to_file(lines,
337                 ↪ os.path.basename(input_string).split('.')[0]
338                 ↪ +'_glosas/glosa'+str(str(a-1).zfill(3)+'_ttml'),
339                 ↪ encoding='utf-8')
340
341         print('\n')
342
343         #print(soup)
344
345     if __name__ == '__main__':
346         if len(sys.argv) != 2:
347             print("Usage: python script.py input_file.ttml")
348             sys.exit(1)
349         main()

```

Script para exibição de múltiplos vídeos simultaneamente:

```

1
2 import subprocess
3 import platform
4 import argparse
5 import threading
6
7 def play_video(player_path, video_path):
8     subprocess.Popen([player_path, video_path])
9
10 def main():
11     # Configurar o analisador de argumentos

```

```
12 parser = argparse.ArgumentParser(description='Reproduzir vídeos
↳ simultaneamente usando MPC-HC no Windows e VLC no Ubuntu')
13 parser.add_argument('videos', metavar='V', type=str, nargs='+',
↳ help='caminhos para os arquivos de vídeo')
14
15 # Analisar os argumentos de linha de comando
16 args = parser.parse_args()
17
18 # Lista de caminhos para os arquivos de vídeo
19 video_paths = args.videos
20
21 # Detectar o sistema operacional
22 os_name = platform.system()
23
24 # Lista para armazenar as threads
25 threads = []
26
27 if os_name == "Windows":
28     # Caminho para o executável do MPC-HC no Windows
29     mpc_hc_path = r'C:\Program Files (x86)\K-Lite Codec
↳ Pack\MPC-HC64\mpc-hc64.exe'
30     # Criar uma thread para cada vídeo usando MPC-HC
31     for video_path in video_paths:
32         thread = threading.Thread(target=play_video,
↳ args=(mpc_hc_path, video_path))
33         threads.append(thread)
34         thread.start()
35 elif os_name == "Linux":
36     # Caminho para o executável do VLC no Ubuntu
37     vlc_path = 'vlc'
38     # Criar uma thread para cada vídeo usando VLC
39     for video_path in video_paths:
40         thread = threading.Thread(target=play_video, args=(vlc_path,
↳ video_path))
41         threads.append(thread)
42         thread.start()
43 else:
44     raise RuntimeError("Sistema operacional não suportado")
45
46 # Aguardar que todas as threads terminem
47 for thread in threads:
48     thread.join()
49
50 if __name__ == "__main__":
51     main()
```